PROGRAMMAZIONE - STORIA - INTERVISTE - RECENSIONI - APPROFONDIMENTI

RETROMAGAZINE WORLD

Future days are back

NUMERO 47 - APRILE 2024- WWW.RETROMAGAZINE.NET

ISSN 3103-697X

COMMODORE 64 LA STORIA

GODZILLA BALLS

IL NUOVO TITOLO PER MSX CHE CI HA CONQUISTATO

PUBBLICAZIONE GRATUITA

Abbiamo fatto 300... mila!

Come sapete non ci piace troppo sbandierare i traguardi raggiunti da RetroMagazine World, ma per questa volta vogliamo fare un'eccezione.
Era da un po' di tempo che facevamo la posta al numero di download e, finalmente, il momento è giunto.
I numeri di RetroMagazine World in italiano ed in inglese sono stati scaricati in totale più di 300.000 volte.

Certo, se guardiamo al numero di visualizzazioni dei video su YouTube, dove alcuni di essi hanno ampiamente passato il miliardo, questo numero fa sorridere, ma per noi di RetroMagazine World e per me personalmente, si tratta di un traguardo notevole che meritava di essere celebrato. Quando nel 2017 abbiamo cominciato questa avventura in pochi credevano che avremmo avuto la costanza di portare avanti un progetto di questa portata. E invece, questi numeri stanno a confermare quanto di buono fatto sinora e quanto ancora ci sia da lavorare. Eh sì, perché anche se 300.000 download possono sembrare tanti, sono soltanto una pietra miliare nel nostro percorso a cui auguriamo ben altri traguardi.

Ah, poi c'è da aggiungere un'altra considerazione...
Questi sono soltanto i download che possiamo contare dal nostro sito web, ma dato che alcuni siti fanno il mirroring dei nostri numeri, la somma dei download reali è decisamente piú alta. Quanto più alta non ci è dato saperlo, quindi, per correttezza, noi facciamo sempre e solo riferimento a questo dato.

Ovviamente il nostro intento non è quello di fare numeri, altrimenti dovremmo e potremmo adottare delle strategie diverse, ma quello di proporre ad ogni uscita un numero ricco di contenuti di qualità e della passione che ormai credo tutti i nostri affezionati lettori hanno imparato a conoscere. Però il conforto che questa nostra passione non sia fine a se stessa, ci dà la forza per continuare a lavorare alacremente e per continuare questa magnifica avventura.
Appuntamento allora al traguardo dei 500.000 download, che speriamo di doppiare presto.

Ah, quasi dimenticavo, un immenso GRAZIE a tutti voi Lettori, senza i quali RetroMagazine World sarebbe soltanto un bellissimo esercizio di stile.

Francesco Fiorentini

SOMMARIO

◊ Commodore 64 — Pag. 3
◊ Casi di identità... - parte seconda — Pag. 13
◊ Il gioco del "Mini-Lotto" — Pag. 18
◊ Guida al SGDK - parte seconda — Pag. 24
◊ PunyInform: installazione degli strumenti e compilazione di un gioco — Pag. 28
◊ Il Fortran sul Commodore 64 - il comando MEM — Pag. 32
◊ Invito a ... “Personaggio” per C64 — Pag. 35
◊ ROTOR RAMPAGE — Pag. 36
◊ Intervista a Richard/TND, un maniaco del codice — Pag. 40
◊ Giappone 24^ puntata: Gacha...pon! — Pag. 50
◊ Pang+ (Amstrad GX4000) — Pag. 52
◊ Shadow Gangs (Dreamcast) — Pag. 54
◊ Great Giana Sisters (Plus4) — Pag. 56
◊ King of Demons (SNes) — Pag. 58
◊ Go For It! Goemon (PS1) — Pag. 60
◊ Alice Sisters (Atari Falcon) — Pag. 62
◊ Tetris (Atari 8bit) — Pag. 64
◊ Xiaomei and the Flame Dragon's Fist (MD) — Pag. 66
◊ Harhargon (C64) — Pag. 68
◊ Trucho (MSX) — Pag. 56
◊ Good Knight (C64) — Pag. 72
◊ Savage Princess 2 (ZX) — Pag. 74
◊ Godzilla Balls (MSX) — Pag. 76
◊ Mai Nurse (GB/PC Engine) — Pag. 78
◊ Bomb Jack the Remake (SMS) — Pag. 79
◊ Arkon Blazet (C64) — Pag. 80
◊ Beach Head (C64) — Pag. 82

Hanno collaborato alla stesura di questo numero di RetroMagazine World (in ordine sparso):

- David La Monaca
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Francesco Fiorentini
- Francesco Donatini
- Roberto Del Mar Pirazzini
- Giampaolo Moraschi
- Leonardo Miliani
- Fredrik Ramsberg
- Ingrid Poggiali
- Eugenio Rapella
- Fredrik Ramsberg
- Marta Rossmann
- Gianluca Girelli
- Marco Pistorio
- Michele Ugolini

- Immagine di copertina: **Giuseppe Mangini**
- Layout di copertina: **Carlo N. Del Mar Pirazzini**

Commodore 64

di Leonardo Miliani

Nel 1981 inizia lo sviluppo di un nuovo sistema pensato esclusivamente per il gioco che, dopo alterne vicende, viene riconvertito in un computer vero e proprio. Questo computer viene messo in commercio per sostituire il suo affermato predecessore: non solo fa meglio, diviene esso stesso un best-seller registrando numeri sensazionali e divenendo il computer che ha venduto come nessun altro. Stiamo parlando del Commodore 64, confidenzialmente noto come C64 (fig. 1).

Fig. 1 - Commodore 64
(autore: Evan-Amos – fonte: Wikimedia Commons)

La Commodore prima del C64

Jack Tramiel, un sopravvissuto polacco dei campi di sterminio nazisti emigrato negli Stati Uniti dopo il secondo conflitto mondiale, fonda nel 1954, insieme a Manfred Kapp, la Commodore Business Machines (CBM). I due si sono conosciuti mentre lavoravano insieme alla Ace Typewriter Repair Company, una ditta che ripara macchine da scrivere: dopo qualche anno di lavoro come dipendenti, decidono di mettersi in proprio riparando e vendendo macchine da scrivere usate o ricondizionate. Con Tramiel rimasto solo al comando dell'azienda, dopo alterne vicende la Commodore entra nel mercato delle calcolatrici e successivamente, a metà degli anni '70 del XX secolo, in quello dei computer, realizzando uno dei tre membri di quella che è nota come la "trinità del 1977", il PET: questo si contende il nascente mercato dei personal computer insieme al TRS-80 ed all'Apple II. Le vendite vanno bene: il PET conquista il gradino più alto del podio grazie alla varietà dell'offerta che soddisfa non solo l'utenza domestica ma anche la piccola azienda per via della possibilità di avere anche modelli con schermi ad 80 colonne.

Fig. 2 - Il Commodore VIC-20
(autore: Evan Amos - fonte: Wikimedia Commons)

Nel 1979, però, accade un fatto strano: compare sul mercato VisiCalc, il primo foglio di calcolo elettronico (spreadsheet) per computer, pubblicato per l'Apple II. Il programma diventa un successo istantaneo: tutte le aziende lo vogliono per semplificare la gestione dei propri conti. Essendo un programma inizialmente disponibile solo per l'Apple II, le persone iniziano a comprare questo computer per poter usare VisiCalc. L'Apple II ha anche un altro punto a suo vantaggio: il colore, che manca al PET che genera immagini monocromatiche. Anche il TRS-80 è monocromatico e, per questo, Tandy-Radio Shark, presenta nel 1980 un nuovo computer capace di generare un'immagine a colori, il TRS-80 Color Computer. Nonostante le non brillanti capacità di questo computer, riesce a vendersi comunque discretamente bene grazie al fatto che, essendo un prodotto di Tandy, è disponibile nella sua catena di negozi Radio Shark, che conta migliaia di punti vendita in tutti gli Stati Uniti. Essendo i prezzi dei componenti hardware dei computer in calo, Tramiel decide di entrare nel settore dei computer domestici (home computer) realizzando una macchina a basso costo, capace di immagini a colori e destinata ad un'utenza inesperta, quindi vendibile non solo nei negozi di elettronica ma anche nei grandi magazzini o nei negozi di giocattoli. Nel 1981 viene commercializzato il VIC-20 (fig. 2), un computer collegabile ad un comune televisore domestico e capace di generare immagini a colori e suoni più complessi rispetto ai bip emessi dai PET dotati di altoparlante interno. Il prezzo di circa 300 dollari ne decreta l'immediato successo: il VIC-20 è il primo computer nella storia dell'informatica a raggiungere il milione di esemplari venduti.

Un nuovo chipset

Come ogni azienda che si rispetti, messo in commercio un prodotto si pensa subito al successivo, e così fa la Commodore. Bisogna però fare i conti con una situazione non certo tranquilla all'interno dell'azienda: nonostante il successo del VIC-20, il PET segna ancora ottimi incassi e Tramiel non sembra deciso a toglierlo dal mercato. Questo crea dello scontento: c'è, infatti, una corrente dissidente di diversi ingegneri e dirigenti che non vede di buon occhio il fatto che l'azienda sia orientata prevalentemente verso le macchine da ufficio come il PET e preme per perseguire la strada aperta con il VIC-20, ossia quella di produrre soprattutto macchine destinate ad un'utenza più domestica. Oltretutto, il creatore del PET, Chuck Peddle, se n'è andato di recente, dopo essersi scontrato direttamente con Jack Tramiel proprio per via del VIC-20: Tramiel, infatti, ha bocciato il progetto TOI (The Other Intellect) che Peddle aveva presentato per farlo diventare il futuro VIC-20, privilegiandone un altro. Andandosene il suo creatore, in molti pensano che al PET sia venuta meno la persona di riferimento nonché quella che poteva portarne avanti lo sviluppo.

Invece il settore commerciale spinge per continuare sulla strada del VIC-20 che sta registrando vendite da record, per cui non ha al momento voglia di scendere da questo carro vincente. Tuttavia, gli ingegneri di MOS Technology sanno bene che questo successo non potrà durare: il VIC-20 è obiettivamente inferiore ad altre proposte in circolazione, come i computer ad 8 bit di Atari, il 400 e l'800. D'accordo con Charles Winterble, capo della MOS Technology, essi decidono di iniziare a sviluppare il suo successore. Winterble ne inizia a parlare ad Al Charpentier, il progettista del chip video VIC che ha dato il nome al VIC-20, ed insieme gettano le basi di un nuovo chipset. Il cuore del nuovo computer si chiamerà 6510 (fig. 3), in pratica un 6502 con una porta di I/O aggiuntiva, mentre per la gestione delle periferiche viene progettato il CIA, una versione avanzata del PIA. Il punto cruciale è il reparto

Fig. 3 - La CPU MOS 6510 (autore: Konstantin Lanzet - fonte: Wikimedia Commons)

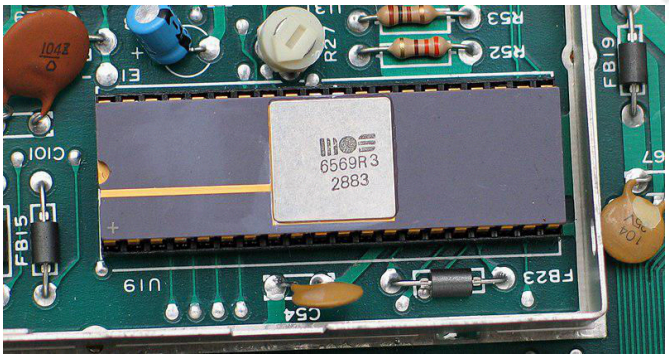


Fig. 4 - Il MOS VIC-II, chip grafico del C64 (foto: Christian Taube - fonte: Wikimedia Commons)

video: il VIC-20 possiede un chip che genera sì un'immagine a colori ma con sole 22 colonne di testo. C'è però nel cassetto un nuovo chip video denominato VIC1.5, che altro non è se non il VIC portato a 40 colonne. Questo chip era stato pensato per una versione a colori del PET nonché come aggiornamento del VIC-20 stesso, pensato per essere sostituito al VIC grazie al fatto che mantiene la piedinatura del chip originale: nessuno dei due progetti ha, però, mai visto la luce, ed il VIC1.5 è rimasto nel cassetto. Winterble chiede un incontro con Tramiel e gli prospetta la volontà di usare questo chip grafico come base per realizzare un nuovo progetto orientato principalmente ai giochi ma trasformabile anche in un computer. A Tramiel piace l'idea e dà l'OK a Winterble, che incarica perciò Charpentier e Bob Russel, un altro ingegnere che ha lavorato sul VIC-20, di studiare i sistemi da gioco in circolazione per prendere degli spunti: questi ultimi, in dieci giorni, esaminano le console ed i computer presenti sul mercato, come l'Intellivision, i computer Atari ed il TI99/4A, e stilano l'elenco delle caratteristiche che il chip deve avere. Alla fine, incontrano nuovamente Tramiel per presentargli il progetto finale, il quale lo approva in via definitiva.

Charpentier si mette al lavoro e progetta una versione migliorata del VIC1.5, che viene chiamato VIC-II: il nuovo chip è capace di generare un'immagine a 40 colonne di testo con una risoluzione di 320x200 pixel e 2 colori su schermo da una tavolozza di 16. Decide però di permettere anche una modalità multicolore con 4 colori differenti per far sì che i giochi siano più realistici, al costo però di dimezzare la risoluzione a 160x200 pixel. Aggiunge poi la gestione di 8 sprite in hardware con tutta una serie di caratteristiche avanzate come il controllo delle collisioni e la possibilità di gestirli anche multicolore: gli sprite sono una cosa così importante che ben 2/3 della superficie del chip sono dedicati alla loro gestione. Viene poi mantenuto lo scrolling hardware in orizzontale e verticale,

utile per i giochi con la scena in movimento. Per il suono viene deciso di creare un chip audio separato, abbandonando quindi la strada del sistema integrato seguita per il VIC (che genera immagine e audio) e seguendo quella già battuta dalla concorrenza. A questo progetto si dedica Bob Yannes, un amante della musica e dei sintetizzatori digitali, il quale realizza un chip con capacità audio mai viste prima su un sistema dometistico: il MOS SID (Sound Interface Device) è capace di generare suoni usando 4 forme d'onda a più ottave. Siccome lo sviluppo porta via molto tempo e Yannes non riesce a concretizzare tutte le idee che ha in mente, al posto di multiplexare l'oscillatore audio e creare più voci da un'unica sorgente interna, decide di replicare 3 volte il circuito creato: è così che il SID è dotato di 3 generatori di suono indipendenti.

Verso il Commodore 64

Nel frattempo, Tramiel ha deciso che il progetto del computer sia grandioso: l'erede del VIC-20 dovrà essere non un computer qualunque, bensì il più potente sistema della sua categoria. Jack, sicuro del fatto che i prezzi delle memorie scenderanno nel periodo che intercorre fra l'inizio della progettazione e la messa in vendita, decide che il nuovo computer venga dotato di ben 64 KB di RAM. Parte lo sviluppo di quello che inizialmente è il progetto VIC-40, un nome scelto non a caso: con esso si vuole far intendere che questa macchina è il successore del VIC-20. Tutti i suddetti chip sono consegnati da MOS Technology per il mese di novembre del 1981, ed il team inizia subito un vero e proprio "tour de force", dato che Charpentier e Tramiel hanno deciso di presentare la macchina al Consumer Electronic Show (CES) di gennaio del 1982. Nonostante saltino anche le festività del Giorno del Ringraziamento e quelle di Natale e lavorino alle volte anche 16/20 ore al giorno, gli ingegneri riescono a completare il prototipo solo pochi giorni prima del CES, la scadenza prefissata.

Poco prima della presentazione ufficiale, il nuovo computer viene ribattezzato semplicemente "C64", dal quantitativo di memoria installata. Il C64 è il vero mattatore del CES e lo stand della Commodore è letteralmente preso d'assalto dagli operatori del settore per assistere alla presentazione del computer: colpiscono non solo le strabilianti caratteristiche tecniche, con un suono ed una grafica mai visti prima su un home computer, ma anche e soprattutto il fatto che costi solo 595 dollari. Terminato l'evento, il C64 torna sui banchi degli ingegneri: il prototipo deve essere convertito in un computer che può essere prodotto. Ci sono ancora molti problemi da risolvere, alcuni derivanti dal chip video, che mostra artefatti grafici in alcune condizioni, altri causati dai problemi di velocità di lettura e scrittura del nuovo lettore di dischi; infine, c'è il sistema operativo, che va adattato al nuovo hardware. Tutto questo lavoro porta via molto tempo: il C64, che doveva debuttare a primavera, viene alla fine messo in commercio solo nel mese di agosto del 1982.

Il MAX (o MAX Machine)

Nel frattempo, Tramiel vuole monetizzare il lavoro fatto e decide di mandare in produzione il progetto parallelo basato sullo stesso chipset del C64 ma destinato a diventare una console, progetto noto in azienda come MAX (fig. 5). Il MAX deve replicare in Giappone il successo del VIC-20 avuto sul suolo americano, combattendo contro i prodotti del Sol Levante sul loro stesso terreno: è perciò un progetto per una console da giochi da vendere intorno ai 180 dollari. Dato che usa gli stessi componenti del C64, le cartucce dei giochi di questi due sistemi vengono rese intercambiabili: per far ciò, il C64 viene modificato creando una modalità che rende compatibile la mappa della memoria del computer con quella della console così che, all'inserimento della cartuccia, il C64 "diventi" un MAX. Lo sviluppo iniziale avviene negli USA ma la finalizzazione dello stesso viene fatta in Giappone presso la filiale locale della Commodore da Yash Terakura. Sebbene nasca come console da gioco, il MAX è stato pensato per diventare anche un piccolo computer, una soluzione molto in voga all'epoca ed adottata anche da altri produttori di sistemi da gioco, come il ColecoVision trasformabile nel computer Adam, il kit Spectravideo CompuMate per l'Atari 2600, il Keyboard Component per il Mattel Intellivision, e altri ancora. Ma invece di vendere la tastiera come accessorio, la nuova macchina viene

Fig. 5 - Il Commodore MAX, o MAX Machine (foto: Thomas Conté - fonte: Wikimedia Commons)

inserita direttamente all'interno di un piccolo case con una tastiera a membrana simile a quella del PET. La trasformazione avviene tramite una cartuccia del BASIC che fornisce, oltre all'interprete, anche un editor a schermo simile a quello del C64 per poter scrivere piccoli programmi. La console viene messa in commercio nel solo Giappone agli inizi del 1982 come Commodore MAX (è nota anche come MAX Machine). Nonostante l'aspetto di un computer, il MAX è una console a tutti gli effetti: lo dimostrano i soli 2 KB di RAM installati nonché la totale assenza di un sistema operativo da lanciare nel caso in cui non sia presente all'accensione nessuna cartuccia. In seguito si pensa di esportarla anche al di fuori dell'isola nipponica: viene perciò presentata come Ultimax in Canada, sempre come MAX in Gran Bretagna e poi come VC-10 in Germania. Il MAX non riesce, però, ad imporsi vista la poca differenza di prezzo con il VIC-20 che, grazie alla sua maggior espandibilità e duttilità, viene preferito alla console. Per questo motivo, viene tolta dal mercato giapponese dopo pochi mesi dall'inizio della sua commercializzazione e qualche decina di cartucce prodotte, e si decide di non metterla neanche in commercio sugli altri mercati dove è stata presentata perché ritenuta inadatta a fronteggiare la concorrenza.

Il debutto del C64 e la guerra dei prezzi

Come detto, il C64 viene messo in commercio ad agosto al prezzo di 595 dollari. Tutta la concorrenza si chiede come possa una macchina del genere costare così poco. La risposta è: "risparmio e riciclo". Per il C64 tutto è stato progettato per poter essere riciclato da progetti precedenti o sviluppato ex-novo col minimo investimento. Il chipset, l'unica cosa quasi del tutto completamente nuova rispetto ai computer già prodotti, è stato comunque sviluppato in casa dalla sussidiaria MOS Technology. Il KERNAL del computer è quello del VIC-20, riciclato pari pari e soltanto adattato al nuovo hardware. Pure il BASIC è lo stesso: difatti la versione è proprio la 2.0 del suo predecessore. Anche il contenitore è identico a quello del VIC-20, solo di un colore differente, quindi costo di progettazione pari a zero.

Con questo prezzo di vendita, che comprende grafica e sprite a colori, capacità audio pari a quelle di un sintetizzatore digitale, un quantitativo di memoria normalmente presente su macchine molto più costose, e con la possibilità di collegare periferiche di ogni genere e la comodità di utilizzarlo anche su un comune TV domestico senza dover acquistare monitor appositi, il C64 sbaraglia la concorrenza.

Fig. 6 - Il Texas Instruments TI-99/4A, una delle vittime della guerra dei prezzi mossa da Jack Tramiel (foto: Abraham San Pedro Salazar - fonte: Wikimedia Commons)

Dal debutto alla fine di quell'anno il C64 vende bene e registra, dopo le feste di Natale, 360.000 unità vendute. Ma questo è niente rispetto ai numeri che farà segnare in seguito... Nel 1983, grazie ad una distribuzione che avviene non solo nei negozi specializzati ma anche in grandi catene commerciali e persino nei negozi di giocattoli, il C64 raggiunge una platea di clienti molto superiore rispetto alle macchine della concorrenza, vendute solo nei negozi specializzati: le persone lo acquistano invogliati dalla stampa di settore che ne elogia l'ottimo rapporto qualità/prezzo. In quell'anno si contano circa 1,3 milioni di unità vendute: i computer ad 8 bit di Atari, le serie 400 ed 800, si piazzano al secondo posto, seguiti dall'Apple II. Quest'ultimo viene velocemente surclassato per via del prezzo nettamente superiore, quasi il triplo del prezzo del prodotto Commodore. Da lì inizia una vera e propria guerra dei prezzi, con una campagna commerciale aggressiva da parte di Tramiel. Il motivo è semplice: far fuori i rivali, soprattutto la Texas Instruments. Tramiel, infatti, si ricorda ancora quando la Commodore, agli inizi degli anni '70 del XX secolo, da leader indiscussa del mercato delle calcolatrici ha quasi rischiato la bancarotta a causa dell'ingresso nel settore proprio dell'azienda texana che, da solo produttore di chip, si è messa a vendere le calcolatrici complete dominando il mercato con prezzi bassissimi grazie al fatto che gli integrati che vende alla concorrenza costano più di una propria calcolatrice. Prima della fine dell'anno il VIC-20 viene ribassato sotto alla soglia dei 200 dollari, risultando più economico del suo diretto rivale, il TI-99/4A della Texas (fig. 6): in risposta, quest'ultima offre uno sconto di 100 dollari sul prezzo di acquisto del computer, che è di 300 dollari. Contemporaneamente Atari ribassa il suo 800 a 500

dollari per contrastare le vendite del C64 del periodo natalizio. Come riposta, agli inizi del 1983, Commodore ribassa il C64 da 595 a 399 dollari, senza però rimetterci: questo è possibile grazie al fatto che l'azienda produce i chipset dei propri computer.

A febbraio, Texas Instruments ribassa il TI-99/4A a 150 dollari: come risposta, ad aprile Commodore taglia i prezzi di tutta la linea del VIC-20, il cui prezzo scende sotto ai 100 dollari. Non solo, attua anche una campagna promozionale che letteralmente demolisce la concorrenza: qualsiasi computer reso alla Commodore darà diritto ad un rimborso di 100 dollari sull'acquisto del C64. Questa promozione non solo riempie i magazzini della Commodore di computer di qualsiasi tipo ma, per assurdo, incrementa, anche se temporaneamente, le vendite dei microcomputer della Timex Sinclair, che importa negli USA i prodotti della Sinclair: la gente acquista economici Timex-Sinclair a meno di 50 dollari per spedirli alla Commodore, ottenere il rimborso di 100 dollari, e intascarsi la differenza.

A maggio la Atari ribassa l'Atari 800 di 100 dollari ma, al CES di giugno, la Commodore annuncia che il C64 passa a soli 200€, un terzo del prezzo iniziale! Ad ottobre la Texas Instruments issa bandiera bianca e si arrende alla Commodore, ritirando il suo TI-99/4A dal mercato: il nemico di Tramiel esce sconfitto. Atari, invece, subisce un colpo ancor più pesante: complice anche la cosiddetta crisi dei videogiochi del 1983 che colpisce il mercato nordamericano dei videogiochi, alcune scellerate scelte di mercato prese per la sua console 2600 si rivelano degli enormi fiaschi commerciali, ed Atari viene trascinata nel baratro. L'azienda registra più di 500 milioni di dollari di perdite: Atari per poco non ci lascia le penne, e solo una forte riorganizzazione societaria, con lo smembramento in due entità distinte dei reparti arcade e sistemi domestici e la vendita di quest'ultimo, la salva dalla bancarotta.

A questo punto, l'ascesa del C64 continua senza rivali. Anche la Apple, nonostante gli aggiornamenti al suo Apple II, non rappresenta più una minaccia dato che il costo di questo computer è molto superiore a quello del C64. Il 1984 vede quest'ultimo registrare 2,6 milioni di unità vendute, il picco più alto mai raggiunto durante tutto il periodo in cui resta in vendita: successivamente, le vendite proseguono su circa 1,5 milioni di unità all'anno fino al 1989, quando calano a circa 1 milione di unità all'anno. Il C64 resiste anche al "fuoco amico": nel 1985, la Commodore presenta il suo successore, il C128. Nonostante caratteristiche più avanzate e la quasi completa compatibilità con il C64, la domanda per quest'ultimo resta sempre forte tanto che il C128 viene tolto dal mercato dopo 4 anni con circa 2,5 milioni di unità vendute, meno di quanto il C64 abbia fatto registrare nel suddetto anno d'oro del 1984. Nello stesso 1985 viene presentata anche l'Amiga: sebbene sia un computer a 16 bit con caratteristiche enormemente superiori, il C64 regge l'urto anche contro quest'ultimo avversario, grazie anche al fatto che nel 1985 il C64 si può acquistare per 149 dollari ed è quindi una valida alternativa economica per chi vuole un computer performante ma non troppo costoso.

Nel 1992, Commodore afferma di aver venduto circa 12 milioni di C64 dall'inizio della sua commercializzazione fino al 31 dicembre 1991. Nel 1992 le vendite sono di 650.000 unità. Nel 1993 si parla di poco più di 200.000 unità mentre per il 1994 i numeri sono incerti, anche perché questo è l'anno del fallimento di Commodore, per cui è lecito stimare vendite esigue. Possiamo affermare che in tutta la sua vita commerciale il C64 abbia venduto sulle 13 milioni di unità, anche se, secondo altre fonti, i numeri potrebbero essere superiori: in ogni caso, il C64 è il computer che detiene il record di vendite, e difficilmente verrà superato in questo primato.

Caratteristiche tecniche

Il cuore del computer è il MOS 6510, una CPU derivata dal 6502 rispetto al quale presenta una riorganizzazione dei piedini grazie a cui è stato possibile aggiungere una porta di I/O a 8 linee (nel C64 ne sono usate solo 6) utilizzata per controllare la mappatura della memoria del computer nonché per comandare direttamente l'eventuale lettore a cassette collegato. A parte questa differenza esterna, il 6510 è in pratica un 6502 per cui il software può essere adattato semplicemente modificando le parti che accedono all'hardware della macchina.

Rispetto al VIC-20, dove un unico coprocessore si occupa della generazione del segnale video e dell'audio, sul C64 questi compiti vengono divisi fra 2 chip appositi: il VIC-II si occupa dell'immagine mentre il SID si occupa del suono. Il VIC-II è un chip capace di generare un'immagine a colori sia in modalità grafica che testuale: nella prima, la schermata è composta da 320x200 pixel mentre nella seconda da 40x25 caratteri, ed in entrambe le modalità i colori possibili sono 16. In modalità grafica si può operare sia in alta che in media risoluzione, quest'ultima è detta anche multicolore: nella prima l'immagine è divisa in blocchi di 8x8 pixel ed ogni blocco permette un colore di sfondo, uguale per tutta l'immagine, ed un colore primario diverso. La modalità multicolore è stata prevista dai

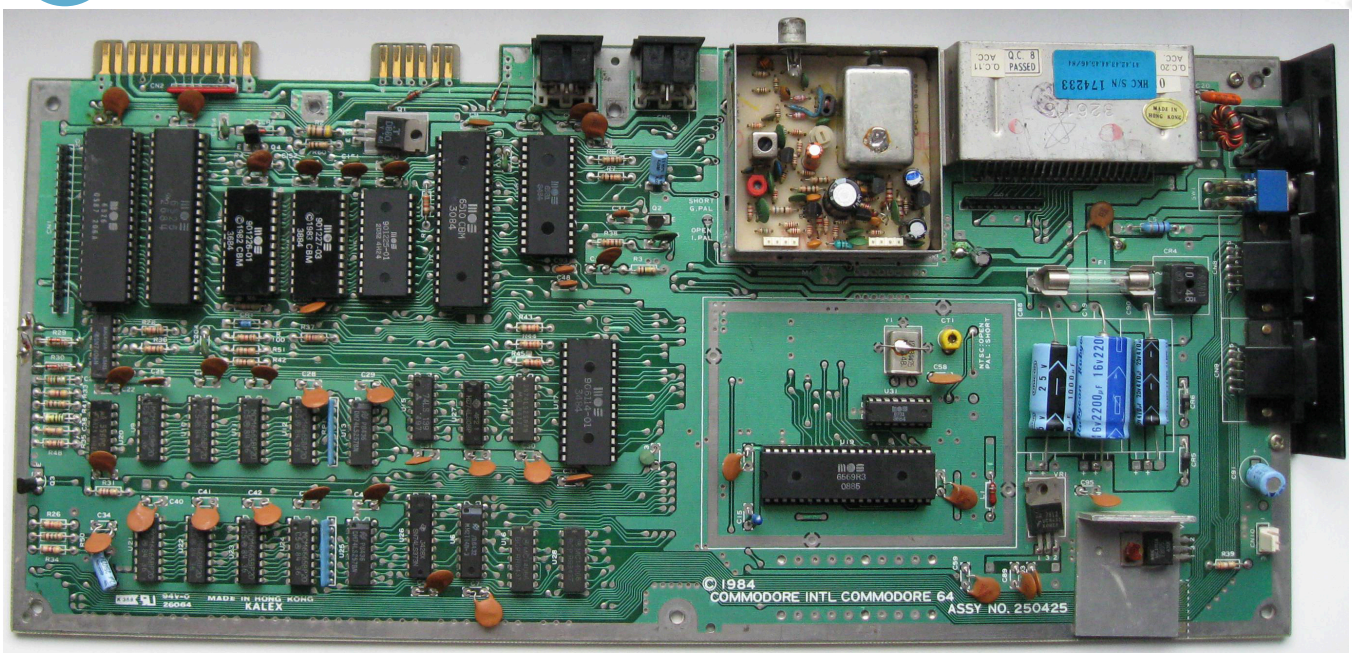


Fig. 7 - Una scheda madre del C64 del 1984: si notano centralmente in alto il MOS SID e, alla sua sinistra, la CPU 6510. In basso il chip posto orizzontalmente è il VIC-II (foto: Gona.eu - fonte: Wikimedia Commons)

progettisti del chip per favorire i programmatori di giochi: in essa, al posto di usare 1 bit per ogni pixel colore ne vengono usati 2. In questo modo si possono indirizzare 4 colori differenti per blocco, al costo però di dimezzare la risoluzione orizzontale, che passa a 160 pixel, e, di conseguenza, i blocchi sono di 4x8 pixel. Le modalità testuali sono 3: la modalità classica, con 1000 celle (40x25) di 8x8 pixel, ognuna delle quali in grado di visualizzare caratteri con 2 colori; la modalità multicolore, in cui la risoluzione è di 4x8 pixel per cella ma col vantaggio di poter indirizzare 4 colori per cella; la modalità multicolore estesa, che permette 5 colori per cella, al costo però di ridurre la tabella dei caratteri da 256 a 64. Nelle modalità multicolore si hanno delle limitazioni: per non occupare molta memoria, viene impostata una serie di colori comuni a tutto lo schermo ed altri specifici per ogni singolo blocco di pixel. Alcuni dati del colore sono memorizzati in un banco di memoria di 1024 nibble (1024 celle da 4 bit) detta Color RAM a cui accede solo il VIC-II.

Oltre a ciò il chip gestisce 8 sprite monocromatici di 24x21 pixel oppure multicolore di 12x21 pixel con 3 colori, con le limitazioni già viste in precedenza (1 solo colore per ogni sprite e 2 comuni a tutti). Il chip gestisce anche le collisioni per cui si può sapere quando uno sprite collide con un altro o con lo sfondo. Gli sprite possono essere anche ingranditi del doppio (non cambia la risoluzione, dato che semplicemente ogni pixel diventa un blocco di 2x2 pixel sullo schermo). Infine, il VIC-II gestisce sia lo scorrimento verticale che orizzontale nonché un interrupt agganciato al pennello video grazie a cui si possono cambiare gli attributi grafici al raggiungimento di una determinata riga dello schermo, ad esempio per modificare il colore del bordo o dell'immagine.

Il comparto sonoro non è da meno. Creato da un vero musicista ed appassionato di musica elettronica, il SID è un vero sintetizzatore digitale capace di generare 4 diverse forme d'onda: a dente di sega, triangolare, quadrata e rumore bianco. E' dotato di 3 voci separate, ognuna capace di miscelare più forme d'onda per crearne di nuove nonché applicare dei filtri o modulare le onde per creare numerosi effetti. Oltre a ciò, ognuno dei 3 oscillatori può gestire l'ADSR, sigla che sta per Attach-Decay-Sustain-Release, vale a dire la gestione del volume nelle 4 fasi del suono, rispettivamente il tempo che ci mette a raggiungere inizialmente il volume massimo e la durata della prima diminuzione, il tempo di mantenimento durante la generazione principale del suono ed infine quello che ci impiega a raggiungere il volume zero una volta interrotta la penultima fase.

Il sistema operativo della macchina occupa 20 KB: 8 KB contengono il BASIC 2.0, lo stesso del predecessore VIC-20, mentre in altri 8 KB è stipato il KERNAL, ossia le routine di gestione delle funzioni base della macchina. Infine, 4 KB sono destinati ad accogliere la mappa caratteri, composta da due diverse tabelle: la prima contiene i classici caratteri ASCII nonché una serie di caratteri semigrafici ereditati dal PET, mentre la seconda la versione "shiftata" della prima, dove le classiche lettere maiuscole sono sostituite dalla loro versione minuscola. Alla ROM si somma una memoria RAM di 64 KB: dato che il processore

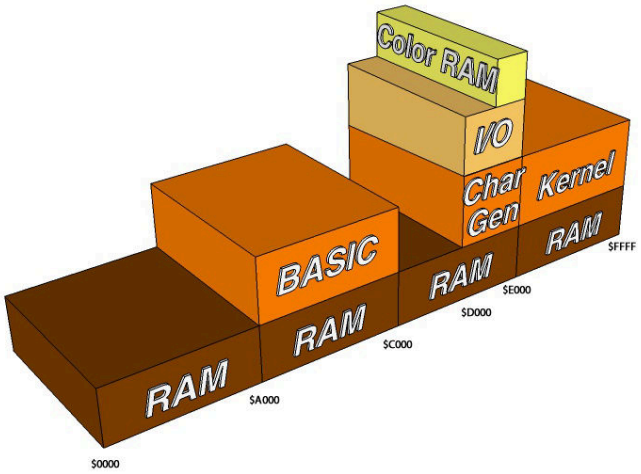


Fig. 8 - La configurazione di memoria del C64. Normalmente la RAM che condivide gli stessi indirizzi della ROM e della memoria di sistema è disattivata (autore: Actraiser - fonte: Dustlayer.com)

6510 non può indirizzare più di 64 KB di memoria, alcuni banchi di RAM sono disattivati ed al loro posto risultano attive le suddette ROM in modo che, normalmente, l'interprete BASIC risulti attivo ed il computer possa gestire gli input dell'utente dati tramite l'editor di schermo. La gestione dei banchi attivi avviene grazie ad un sofisticato sistema di scambio delle memorie ("bank switching") gestito da un apposito integrato, detto PLA. La sintesi di tutto questo si traduce in circa 38 KB di RAM disponibili all'utente per i programmi in BASIC: nel caso in cui si voglia disporre di tutta la memoria disponibile, è possibile disattivare la ROM del sistema operativo ed attivare la RAM sottostante, che occupa gli stessi indirizzi, a costo di rinunciare al BASIC ed al KERNAL. (fig. 8).

Sul lato dell'interfacciamento con il mondo esterno, il C64 offre una grande varietà di possibilità. Dietro al computer c'è un connettore a 44 pin collegato direttamente ai bus dati ed indirizzi della CPU, usato per collegare cartucce giochi su ROM o espansioni di memoria RAM. C'è poi un altro connettore con 24 linee, detto porta utente, al quale si possono agganciare periferiche di terze parti come stampanti o altro. A fianco di questi due connettori primari troviamo un classico connettore per l'uscita RF per collegare un comune apparecchio TV, un connettore DIN ad 8 pin per collegare monitor compatibili con il segnale video composito, un connettore seriale proprietario a cui si possono collegare stampanti e lettori di dischi, un altro connettore a pettine per il collegamento del Datassette (l'unità a nastri di Commodore) ed infine un paio di porte a 9 pin per collegare altrettanti joystick ma anche mouse e penne ottiche.

Versioni ed espansioni

Nel corso della sua lunga vita commerciale (ben 12 anni di carriera!) il Commodore 64 è andato incontro a numerose revisioni e cambiamenti, non solo estetici. Partendo dal comparto tecnico, c'è da segnalare che la Commodore, nell'ottica di ottimizzare la produzione della macchina e rincorrere una perenne riduzione dei costi, ha continuamente aggiornato la scheda elettronica del computer (fig. 7). Questo si riflette sulla dimensione e forma della scheda ospitata all'interno del computer ma anche sugli integrati stessi: ad esempio, nel corso del tempo sono cambiati il numero dei chip di RAM, la disposizione degli integrati nonché il tipo stesso di chipset montato (quando ad esempio sono uscite le nuove versioni con processo produttivo aggiornato).

A parte l'elettronica, ci sono state anche delle versioni a volte che hanno riguardato l'estetica o le capacità del C64. La prima versione speciale è stata la console MAX, o Ultimax com'era nota all'interno dell'azienda, messa in vendita agli inizi del 1982, prima addirittura del C64 stesso ma durata veramente poco tempo. L'Educator 64 è stato un tentativo fatto dalla Commodore nel 1983 di erodere le quote dell'Apple II negli istituti scolastici. Rilasciato con un case simile a quello del PET e con un monitor integrato a fosfori verdi, il computer non ha avuto fortuna dovendo battersi contro l'esteso parco software del primo. Sempre nel 1983 viene presentato il Commodore SX-64 (fig. 9), un C64 che riprende lo stile dell'Osborne 1 e del Kaypro II, ossia quello di un computer "portabile"

Fig. 9 - Il Commodore SX-64, una versione "portabile" del C64 (foto: Hedning - fonte: Wikimedia Commons)

Fig. 10 - Il Commodore 64C, che riprende lo stile del C128 (foto: Kausalkette - fonte: Wikimedia Commons)

delle dimensioni di una piccola valigia, rispetto a questi sistemi è stato il primo ad offrire un monitor a colori.

Nel 1986, seguendo lo stile del C128 introdotto l'anno prima, viene presentato il C64C, una versione con un case più basso e filante e di colore beige chiaro (fig. 10). Funzionalmente identico al precedente modello, alcune periferiche però non si adattano più per via dei diversi ingombri, ad esempio le schede acceleratrici da montare internamente o alcuni accessori esterni come le tastiere musicali da applicare sui tasti della macchina. A parte questo, internamente si segnala il passaggio alla nuova versione del chipset, che porta con sé un problema col SID: per correggere un "difetto" del progetto iniziale, il SID ha perso quella forza e quella "ruvidezza" iniziali nonché alcuni effetti sonori che, nel corso degli anni, diversi musicisti hanno imparato a tirar fuori trasformando un difetto in un pregio. Se ad alcuni il suono più pulito piace di più, gli amanti del vecchio C64 gridano allo scandalo, non sentendo più alcuni suoni che il SID originale era in grado di riprodurre rispetto al nuovo. Verso la fine degli anni '80 del XX secolo la Commodore produce, per il mercato tedesco, un paio di versioni speciali (la "ALDI" e la "G") per le quali viene rispolverato il vecchio case: quindi, "ibridi" con le forme del primo C64 ed i colori delle plastiche più chiari come sul C64C.

Nel 1990, provando (erroneamente) a ripercorrere (malamente) la strada (errata) del MAX, viene rilasciato il Commodore 64 Game System, abbreviato in 64 GS, un C64 privato della tastiera e di tutto tranne la porta per le cartucce e quelle per i joystick. Il C64 GS è in pratica la versione "console" del computer e può far funzionare le sue cartucce giochi a patto che questi non richiedano input da tastiera, dato che non esiste... Inutile dire che anche qui siamo al nuovo, ennesimo, fiasco commerciale.

Infine, c'è da citare il mitico Commodore 65. Creato nel 1991, è una versione a metà strada fra il C64 ad 8 bit e l'Amiga a 16 bit: il progetto è stato poi cancellato, probabilmente perché il mercato stava lasciando anche i 16 bit per orientarsi verso i più performanti 32 bit. Nel 1994, dopo il fallimento della Commodore, alcune decine di prototipi del C65 sono usciti dall'azienda e sono stati messi in vendita, divenendo nel corso degli anni oggetti da collezione molto rari.

Giochi e programmi

Inutile dire che le capacità grafiche e sonore del computer nonché la sua enorme diffusione ha sempre attratto gli sviluppatori software, tanto che a metà degli anni '80 del XX secolo affinché un qualunque programma abbia successo basta che venga pubblicato per il PC IBM e per il Commodore 64. Molti programmi di utilità sono stati convertiti anche per il C64, ad esempio il foglio elettronico MultiPlan di Microsoft, oppure KoalaPainter, il programma di disegno offerto in abbinamento alla tavoletta grafica KoalaPad. Una grossa spinta alla diffusione di programmi "seri" si è avuta dopo l'introduzione del GEOS, un sistema operativo grafico rilasciato da Berkley Software nel 1986: originariamente sviluppato proprio per il C64, il GEOS offre un ambiente grafico che ricorda molto quello del primo Macintosh di Apple e permette, essendo un vero sistema operativo, l'esecuzione di applicazioni che possono non solo accedere a tutte le periferiche della macchina ma avvalersi anche del supporto dell'ambiente a finestre. Oltre a quello integrato, non mancano versioni del BASIC più avanzate che sfruttano meglio le capacità della macchina, ad esempio il Simon's BASIC, che offrono nuovi comandi per l'accesso facilitato ai dischi e per disegnare in modalità grafica, gestire gli sprite o generare musica in maniera più semplice. Rimanendo in tema di linguaggi di programmazione, si è avuta anche un'ampia offerta di interpreti e compilatori per i più diffusi linguaggi dell'epoca, come l'assembly, il Forth, il C, il Pascal, il FORTRAN.

I giochi sono stati, però, quelli che più di tutti hanno mostrato le capacità audio e grafiche del C64: nella seconda metà degli anni '80 più del 60% del software prodotto per questa piattaforma sono giochi per cui è impossibile citarli tutti, sono migliaia. Secondo alcune stime siamo sull'ordine dei 6.000, ed il numero dei programmi complessivamente rilasciati, contando i software applicativi, arriva complessivamente a 10.000. È doveroso poi fare un distinguo fra i giochi pubblicati negli USA e quelli pubblicati in Europa: se i primi sono

Fig. 11 - Impossible Mission
(fonte: Mobygames.com)

Fig. 12 - Winter Games
(fonte: Mobygames.com)

stati rilasciati prevalentemente su dischetto, perché questo era il formato più diffuso in Nord America, in Europa sono stati rilasciati giochi prevalentemente su cassetta, formato più economico e diffuso nel Vecchio Continente. Quindi, molti titoli sviluppati per usare il caricamento di dati in modo random durante il gioco non sono stati pubblicati su nastro mentre non tutti i titoli rilasciati su cassetta, pubblicati prevalentemente da sviluppatori europei, sono arrivati in America. A parte questo, ci sono comunque dei titoli distintivi del computer: "Impossible Mission" del 1984 (fig. 11), dove si vestono i panni di un agente speciale, è uno dei primi giochi a larga diffusione, spesso citato come uno dei giochi "must-have" del C64. Ci sono poi i giochi sportivi multi-evento pubblicati da Epyx, come "Summer Games", "Winter Games" (fig. 12), "California Games" e "World Games". "Pitstop II" va citato perché è il primo titolo di corse automobilistico per 2 giocatori con visione divisa.

Agli inizi degli anni '90, quando ormai la piattaforma è alla fine della sua vita commerciale e quando il mercato è ormai dominato dai sistemi a 16 bit, i programmatori, forti dell'esperienza e della conoscenza della macchina accumulata negli anni precedenti, hanno rilasciato delle vere e proprie perle che hanno sfruttato al massimo le capacità del C64. "Turrican" I (fig. 13) e II sono giochi nati sul C64 rispettivamente nel 1990 e nel 1991 che, data la loro qualità, hanno riscosso così tanto successo da essere stati convertiti per altre piattaforme. "Mayhem in Monsterland" (fig. 14), del 1993, è universalmente noto come il non-plus-ultra dei giochi ad 8 bit: i programmatori sfruttano così a fondo il C64 che il risultato finale, un platform in stile "Supermario", sembra un gioco per un sistema a 16 bit.

Le capacità audio del computer hanno fatto nascere anche un nuovo lavoro, ossia il compositore digitale di musiche per giochi. Tra gli artisti musicali nati e formatisi sul C64 non possiamo non citare Martin Galway, il primo ad usare musica campionata nella sua conversione di "Arkanoid" del 1987, oppure Matt Gray, noto per i famosi temi musicali della serie "The Last Ninja", giochi che, oltre alle musiche, sono da ricordare per le scene isometriche (fig 15). Rob Hubbard ha scritto molte musiche di famosi giochi fra cui

Fig. 13 - Turrican
(fonte: Mobygames.com)

Fig. 14 - Mayhem in Monsterland
(fonte: Mobygames.com)

Fig. 15 - The Last Ninja 2 viene ricordato non solo per la sua visione isometrica ma anche per il suo tema musicale (fonte: Mobygames.com)

Fig. 16 - "We Come in Peace", una demo di Censor Design & Fairlight del 2018 (fonte: Youtube.com)

"International Karate", diversi capitoli della serie "Monty" e "Commando", per citarne alcuni.

Dato che molti giochi sono stati pubblicati con sistemi di protezione, nascono in quell'epoca anche diversi pirati informatici o gruppi di essi che, per vantarsi di aver infranto la protezione di questo o quel software, inseriscono le proprie firme all'inizio del caricamento dello stesso. Queste firme diventano sempre più complesse, per mostrare di cosa essi sono capaci: da semplici testi ben presto si arriva a animazioni grafiche accompagnate da musica. Alla lunga, le animazioni diventano talmente elaborate e sfruttano così a fondo l'hardware della macchina che raggiungono livelli qualitativi inverosimili: dalle "crack-intro", le scene messe all'inizio dei giochi craccati, si passa così al fenomeno delle "demoscene", ossia le crack-intro che diventano programmi indipendenti e che altro non fanno che mostrare le capacità dell'hacker nel portare al limite le macchine su cui girano. Fenomeno che è rimasto attivo per anni e che negli ultimi tempi, complice il rinnovato interesse per i nostri sistemi ad 8 e 16 bit, ha ripreso campo ed è tornato in auge fra gli appassionati di allora ma anche fra i neofiti (fig. 16).

Conclusioni

Potremmo continuare ancora per molto tempo, citando la miriade di periferiche nate per far fare al Commodore 64 qualsiasi tipo di lavoro. Io stesso ricordo che, a metà degli anni '80, andai a far visita ad un amico il cui genitore aveva il computer collegato ad un modem e, tramite un programma, riceveva le immagini dai satelliti Meteosat, le elaborava e mostrava poi sullo schermo la situazione meteo dell'Italia in tempo reale. Miriadi di ragazzi hanno costruito le loro carriere informatiche imparando i rudimenti della programmazione su questo computer, e chi non programmava, ha comunque imparato ad usare i computer grazie al C64. Le edicole erano invase da riviste con programmi da digitare sul proprio computer nonché di raccolte di giochi per provare a casa quel nuovo arcade visto al bar. Sì, perché nonostante le limitazioni del suo BASIC, quando si passava all'assembly e si spremeva la macchina, questa tirava fuori cose che gli altri computer coevi potevano solo immaginarsi. Sappiamo poi tutti come è andata a finire la storia, con la Commodore che non è mai riuscita a sfruttare al meglio tutte quelle opportunità che ha avuto fra le mani, dall'Amiga allo stesso C64, ottenendo una frazione di quanto questi sistemi potenzialmente avrebbero potuto portare in termini economici se solo si fosse pensato a svilupparli ed evolverli in modo intelligente invece che concentrarsi sul prendere la gallina oggi lasciando perdere l'uovo di domani.

Casi di identità... - parte seconda

di Marco Pistorio

Bentrovati nuovamente, amici lettori di RetroMagazine World!

Nella prima parte di questo articolo, pubblicato sul numero 42 di RetroMagazine World, Vi ho parlato di una mia esperienza nello sviluppo di alcuni lavori in assembly per il nostro amato biscottone. Mi sono soffermato sul problema di come individuare in maniera precisa se un programma eseguibile coincide esattamente, byte per byte, con uno creato successivamente, anche a distanza di qualche anno.
Vi ho parlato della possibilità di determinare la firma MD5 del file per riuscire a calcolare la sua esatta corrispondenza con un altro file.

Questa è una strada che, di solito, è agilmente percorribile, o lo dovrebbe essere almeno nella maggior parte dei casi.

Vi suggerisco inoltre di documentare sempre i vostri lavori, utilizzando magari delle note all'interno del codice sorgente stesso, indicando sempre la data esatta della compilazione, il compilatore che avete impiegato, specificandone anche il numero di versione, eventuali tools utilizzati (specificandone i numeri di versione anche per questi) ed infine il sistema operativo che avete utilizzato.

Se vi troverete a ricompilare i sorgenti di questi lavori, vi sarà più facile mettere insieme tutti i 'pezzi' per ottenere lo stesso preciso file oggetto, e ne potrete confrontare la firma MD5 con quella del file oggetto creato in precedenza.

Purtroppo, nonostante tali accortezze, il risultato non è sempre garantito.

Vi ho raccontato di una situazione che mi si è presentata, quando mi sono trovato a ricompilare una mia vecchia intro che ho pubblicato nel 2016 su CSDB e non sono riuscito, nonostante vari tentativi, a riottenere lo stesso file .prg identico a quello in precedenza pubblicato.
E' probabile che, in questa precisa circostanza, il motivo sia da addebitare al fatto che, a suo tempo, adoperai una versione di EXOMIZER che girava su sistemi LINUX.
Ritengo che lo stesso tool di compressione dati che gira su Windows si comporti, internamente, in maniera leggermente diversa e, da qui, il risultato è un file .prg leggermente diverso e quindi una firma MD5 non coincidente.
Vi ho lasciato quindi, alla conclusione della parte precedente di questo articolo, con un quesito ovvero: "Come fare a comprendere se due eseguibili siano riconducibili allo stesso codice sorgente riconducibile, quando le rispettive firme MD5 dei due programmi eseguibili non corrispondono?"

Qualcuno penserà: "se i due programmi eseguibili si comportano allo stesso modo...saranno gli stessi!".
Per Voi è una approssimazione accettabile? Io pensavo invece ad una analisi di livello quasi 'forensico' :D

Come si potrebbe procedere? Allora, innanzitutto è bene precisare che dovreste avere a disposizione l'esatto codice sorgente. Dovreste ricompilarlo e poi analizzarlo opportunamente.

Tale analisi dovrebbe essere fatta a pari condizioni. Mi spiego meglio:

Se il programma originale venisse inizialmente decompresso al suo avvio, l'esame del suo codice dovrebbe avvenire immediatamente dopo la sua decompressione.

Perché? Perché partiamo dall'idea che le firme MD5 dei due files siano diverse e che quindi esiste una differenza tra i due files.
Per osservarli dobbiamo partire dalle stesse condizioni iniziali (decompresso il primo, decompresso anche il secondo).
Dobbiamo cercare di comprendere DOVE differiscono, in quali zone di memoria ed in quale misura.

Qui il discorso si fa sicuramente più tecnico ed inoltre, Vi ricordo, che faccio riferimento in maniera esplicita a lavori realizzati per il Commodore 64.
La tecnica potrebbe essere impiegata anche in altri ambiti, tuttavia gli esempi ed i tools che vi illustrerò si riferiscono al Commodore 64.

È necessario ottenere un dump della memoria del C64 subito dopo che il programma sia stato lanciato e sia stato decompresso.

Come? A tale scopo potreste utilizzare questo ottimo (e gratuito) strumento: c64 65XE NES DEBUGGER (rif. figura 1)

https://sourceforge.net/projects/c64-debugger/files/

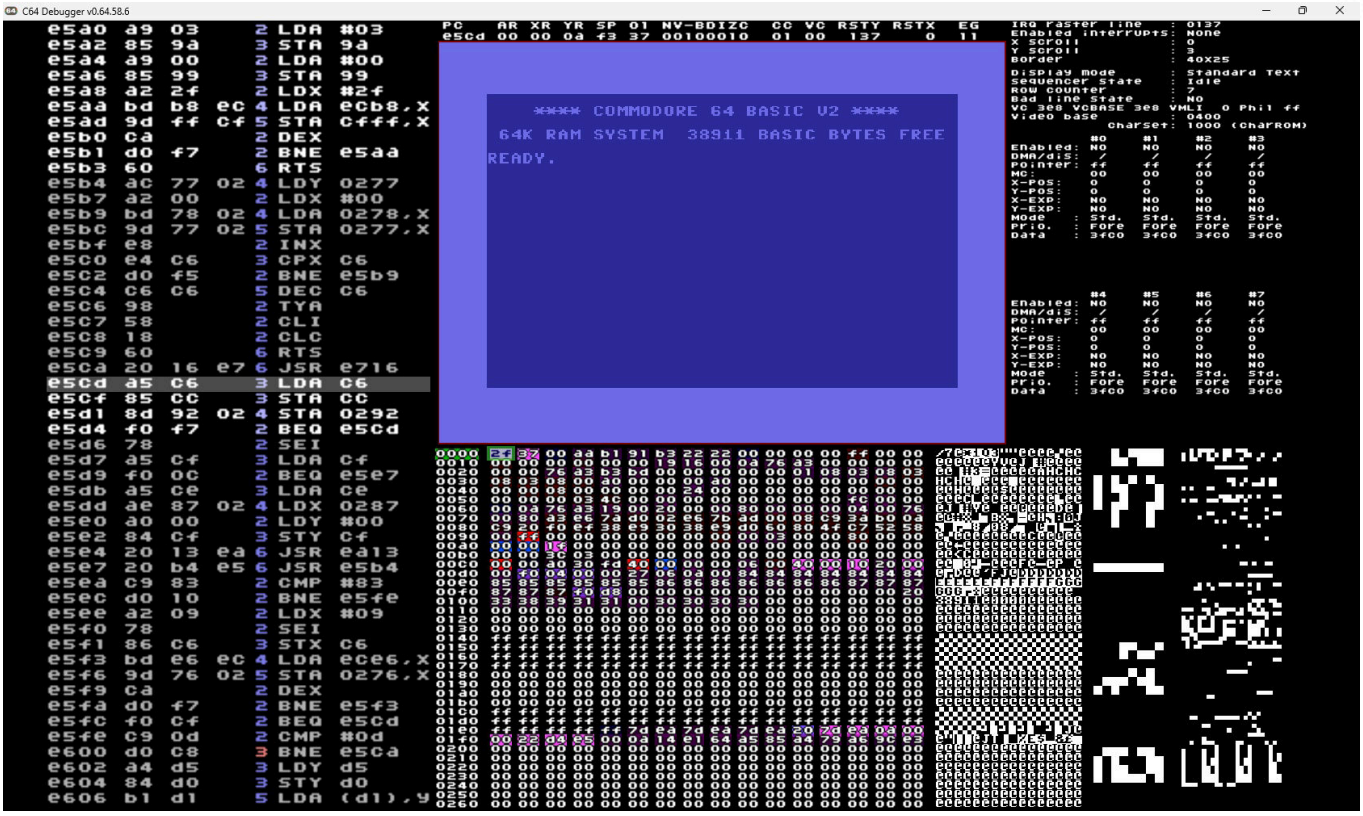


Fig. 1 - C64 Debugger appena lanciato

La procedura per ottenere un dump è la seguente:
1) Lanciate il programma
2) Premete: CTRL+B "BREAKPOINTS"
3) Impostate un indirizzo preciso dove intendete bloccare l'esecuzione del programma,in corrispondenza del riquadro "CPU PC" cioè il registro Program Counter (anche se è possibile sfruttare ulteriori opzioni).
NOTA BENE: QUI vi potrà essere prezioso il codice sorgente del programma per capire DOVE vi conviene impostare il punto di interruzione dal quale poi salvare una immagine completa della memoria del C64.
4) date ancora: CTRL+B
5) Premete CTRL+O e scegliete il file .prg da lanciare. Se avete fatto tutto bene si fermerà l'esecuzione del programma ESATTAMENTE alla locazione che avete indicato nel punto 3.
6) Premete CTRL+U per salvare su file del vostro PC un dump completo della memoria del C64.

Ovviamente, se state confrontando questo primo file .prg con un secondo che avete creato successivamente, dovrete effettuare le stesse operazioni una seconda volta specificando stavolta il nuovo file .prg di cui volete ottenere un dump che poi confronterete.

A questo punto il nostro problema si riduce al confronto tra questi due files di dump. Se sono molto dissimili significa che non possono essere ricondotti al medesimo codice sorgente. Se invece le differenze che rileveremo dovessero essere minime il discorso potrebbe essere diverso.

Anche in questa fase è importante esaminare i sorgenti per determinare quali aree di memoria siano preposte, ad esempio, a contenere dei dati e quali, invece contengano le istruzioni assembly.
Se non conosciamo queste informazioni potremmo non poter stabilire, trovando una differenza tra i due dump, se tale differenza possa essere trascurata o meno.

Il contenuto di ciascun file di dump non è immediatamente leggibile. Conviene utilizzare un decompilatore per renderlo più leggibile.
Io uso spesso un mio personale decompilatore ma ne esistono diversi in circolazione.

Posso segnalarVi, tra i tanti, questo:
Infiltrator Disassembler V1.0 (rif. figura 2)
https://csdb.dk/release/?id=100129

Quando si è riusciti ad elaborare i 2 files .prg, e li si è decompilati, è possibile confrontare insieme i 2 file di testo generati per determinarne le differenze.

Vi segnalo questo ottimo tool online (rif. figura 3):
https://www.diffchecker.com/text-compare/

Finalmente possiamo osservare le differenze, cercando di valutarle.

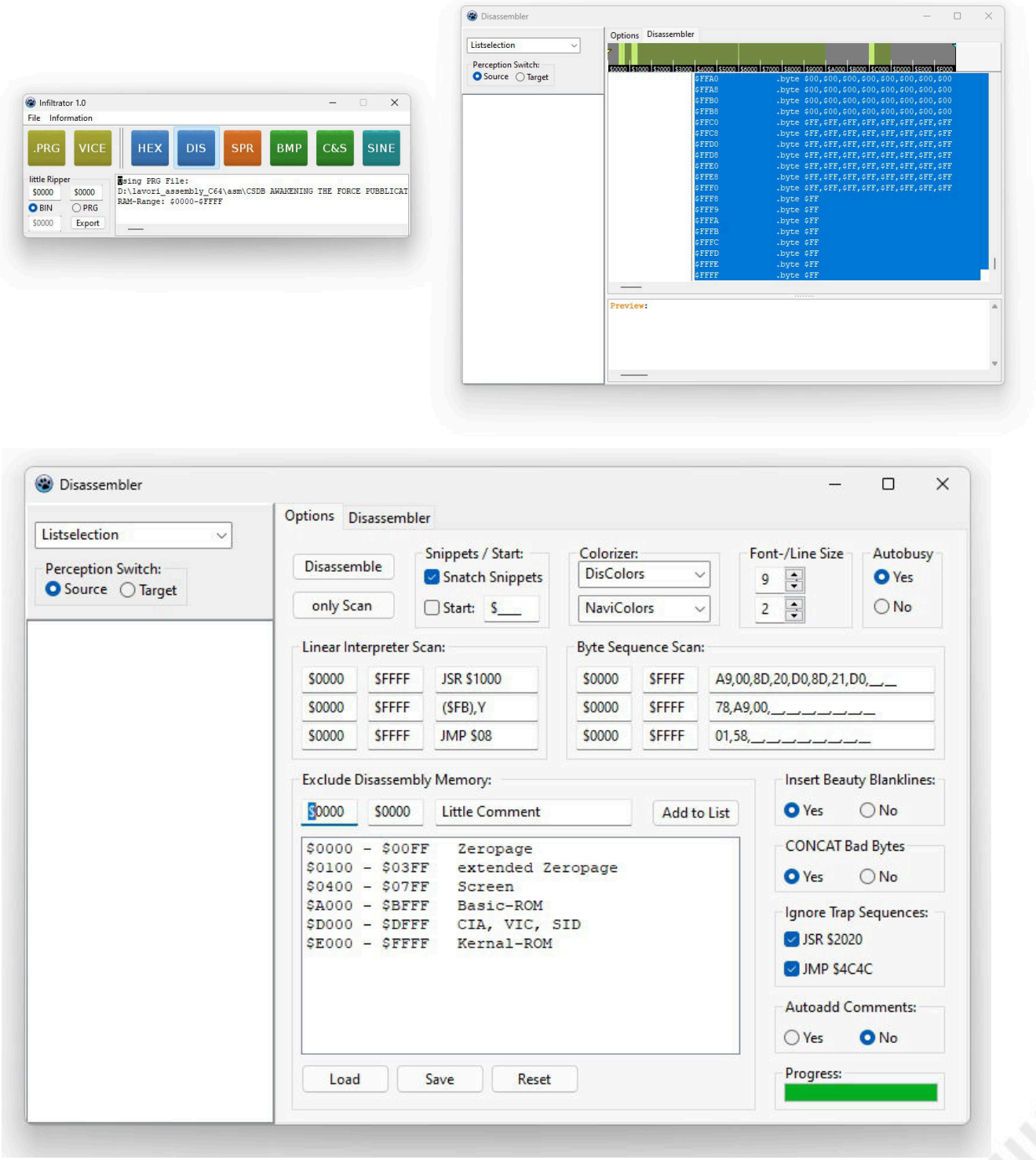


Fig. 2 - Infiltrator Disassembler in azione!

Ho svolto questo lavoro proprio sul dump della mia intro "The Awakening of the Force" che ho pubblicato su CSDB nel 2016 (vedi figura 4) e che ho confrontato con il dump di un file .prg che ottengo ora compilando il medesimo codice sorgente.

Le 2 firme MD5, come già accennato, risultano diverse. Ciò implica che vi siano delle differenze in memoria.

Questa è una lista completa di tutte le differenze che ottengo comparando i due dump tramite miei strumenti:

```
$0023/$0023 :ERR|$0023: A3       |$0023: 08
$0049/$0049 :ERR|$0049: 04       |$0049: 08
$00a1/$00a1 :ERR|$00a1: 12       |$00a1: 13
$00a2/$00a2 :ERR|$00a2: C8       |$00a2: 3D 40 C0
$00a3/$XXXX :ERR|$00a3: 40       |UNAVAILABLE
$00a4/$XXXX :ERR|$00a4: C0 00    |UNAVAILABLE
$00c9/$00c9 :ERR|$00c9: 07       |$00c9: 05 03
$00ca/$XXXX :ERR|$00ca: 03       |UNAVAILABLE
$00cc/$00cc :ERR|$00cc: 01 02    |$00cc: 01 11
$00d1/$00d1 :ERR|$00d1: 80       |$00d1: E0 05
$00d2/$XXXX :ERR|$00d2: 06 00    |UNAVAILABLE
$00d6/$00d6 :ERR|$00d6: 10 0D    |$00d6: 0C
```

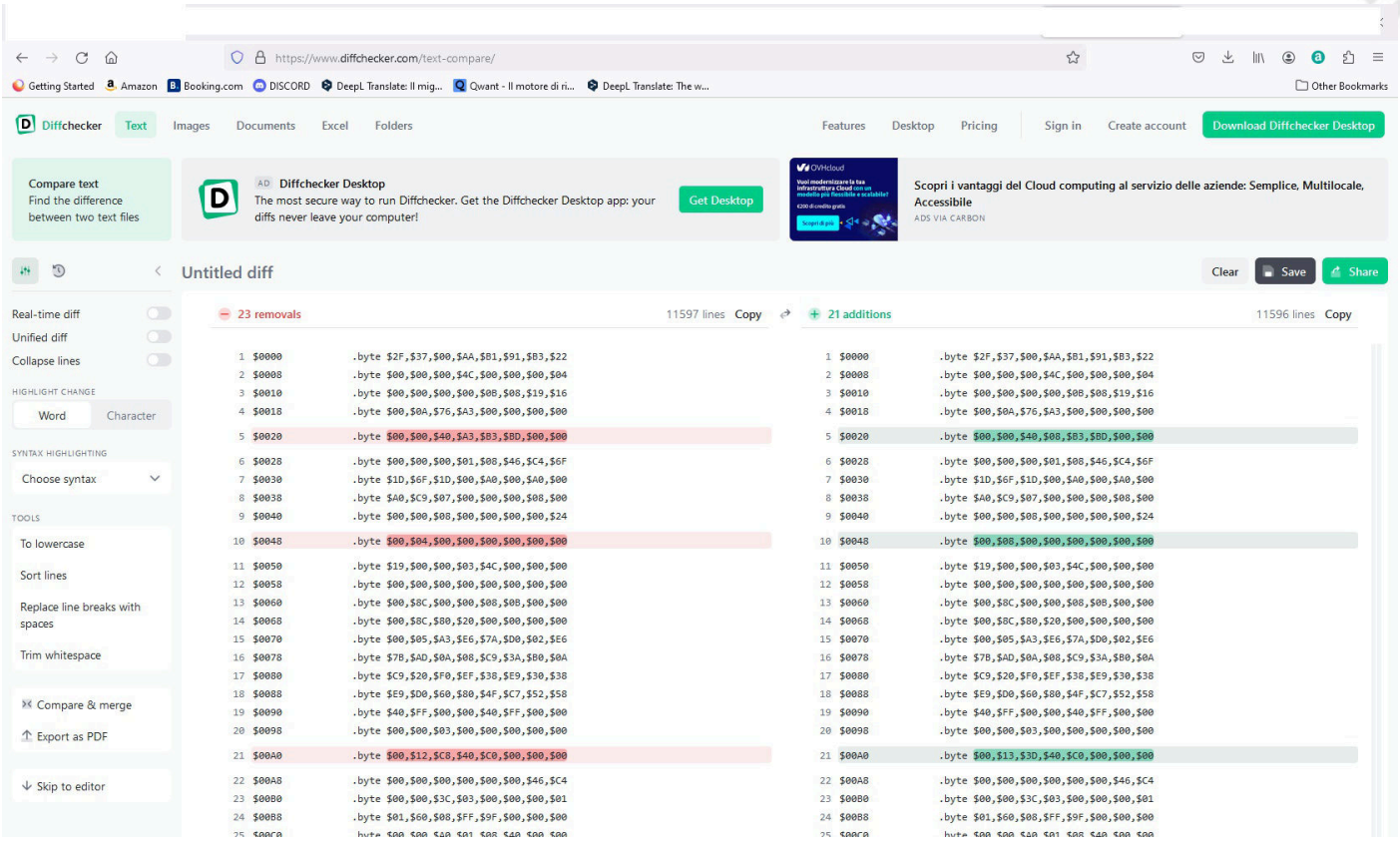


Fig. 3 - Risultato del confronto tra i due files di dump decompilati

```
$00d8/$XXXX :ERR|$00d8: 00       |UNAVAILABLE
$00d9/$XXXX :ERR|$00d9: 84 84    |UNAVAILABLE
$00db/$XXXX :ERR|$00db: 84 84    |UNAVAILABLE
$00dd/$XXXX :ERR|$00dd: 84 84    |UNAVAILABLE
$00df/$XXXX :ERR|$00df: 84 85    |UNAVAILABLE
$00e1/$XXXX :ERR|$00e1: 85 85    |UNAVAILABLE
$00e3/$XXXX :ERR|$00e3: 85 85    |UNAVAILABLE
$00e5/$XXXX :ERR|$00e5: 85 86    |UNAVAILABLE
$00e7/$XXXX :ERR|$00e7: 86 86    |UNAVAILABLE
$00e9/$XXXX :ERR|$00e9: 86 86    |UNAVAILABLE
$00eb/$XXXX :ERR|$00eb: 86 86    |UNAVAILABLE
$00ed/$XXXX :ERR|$00ed: 87       |UNAVAILABLE
$00f3/$00f3 :ERR|$00f3: 80       |$00f3: E0 D9
$00f4/$XXXX :ERR|$00f4: DA       |UNAVAILABLE
$00f5/$00f5 :ERR|$00f5: 81 EB    |$00f5: C2
$01f7/$01f7 :ERR|$01f7: 00       |$01f7: FF
$0202/$0202 :ERR|$0202: 4E 00 00 |$0202: 4E 00 2C
$0286/$0286 :ERR|$0286: 0E 0E 04 |$0286: 0E 6E 04
$028c/$028c :ERR|$028c: 10 00    |$028c: 10 01
$028e/$028e :ERR|$028e: 00       |$028e: 01 48
$028f/$XXXX :ERR|$028f: 48       |UNAVAILABLE
$05b8/$05b8 :ERR|$05b8: 0C       |$05b8: 12
$05b9/$05b9 :ERR|$05b9: 09 13    |$05b9: 15 0E
$05bb/$05bb :ERR|$05bb: 14       |$05bb: 20 20 20
$05bc/$XXXX :ERR|$05bc: 20 20 20 |UNAVAILABLE
$05bf/$XXXX :ERR|$05bf: 20 20 20 |UNAVAILABLE
$05c2/$XXXX :ERR|$05c2: 20 20 20 |UNAVAILABLE
$05c5/$XXXX :ERR|$05c5: 20 20 20 |UNAVAILABLE
$05c8/$XXXX :ERR|$05c8: 20 20 20 |UNAVAILABLE
$05cb/$XXXX :ERR|$05cb: 20 20 20 |UNAVAILABLE
$05ce/$XXXX :ERR|$05ce: 20 20 20 |UNAVAILABLE
$05d1/$XXXX :ERR|$05d1: 20 20 20 |UNAVAILABLE
$05d4/$XXXX :ERR|$05d4: 20 20 20 |UNAVAILABLE
$05d7/$XXXX :ERR|$05d7: 20 20 20 |UNAVAILABLE
$05da/$XXXX :ERR|$05da: 20 20 20 |UNAVAILABLE
$05dd/$XXXX :ERR|$05dd: 20 20 20 |UNAVAILABLE
$05e0/$XXXX :ERR|$05e0: 20 20 20 |UNAVAILABLE
$05e3/$XXXX :ERR|$05e3: 20 20 20 |UNAVAILABLE
$05e6/$XXXX :ERR|$05e6: 20 20 20 |UNAVAILABLE
$05e9/$XXXX :ERR|$05e9: 20 20 20 |UNAVAILABLE
$05ec/$XXXX :ERR|$05ec: 20 20 20 |UNAVAILABLE
$05ef/$XXXX :ERR|$05ef: 20 20 20 |UNAVAILABLE
$05f2/$XXXX :ERR|$05f2: 20 20 20 |UNAVAILABLE
$05f5/$XXXX :ERR|$05f5: 20 20 20 |UNAVAILABLE
$05f8/$XXXX :ERR|$05f8: 20 20 20 |UNAVAILABLE
$05fb/$XXXX :ERR|$05fb: 20 20 20 |UNAVAILABLE
$05fe/$XXXX :ERR|$05fe: 20 20 20 |UNAVAILABLE
$0601/$XXXX :ERR|$0601: 20 20 20 |UNAVAILABLE
$0604/$XXXX :ERR|$0604: 20 20 20 |UNAVAILABLE
$0607/$XXXX :ERR|$0607: 20 31 39 |UNAVAILABLE
$060a/$XXXX :ERR|$060a: 39 33 20 |UNAVAILABLE
$060d/$XXXX :ERR|$060d: 13       |UNAVAILABLE
$060e/$XXXX :ERR|$060e: 19 13 32 |UNAVAILABLE
$0611/$XXXX :ERR|$0611: 30 35    |UNAVAILABLE
$0613/$XXXX :ERR|$0613: 39 20 20 |UNAVAILABLE
$0616/$XXXX :ERR|$0616: 20 20 20 |UNAVAILABLE
$0619/$XXXX :ERR|$0619: 20 20 20 |UNAVAILABLE
$061c/$XXXX :ERR|$061c: 20 20 20 |UNAVAILABLE
$061f/$XXXX :ERR|$061f: 20 20 20 |UNAVAILABLE
$0622/$XXXX :ERR|$0622: 20 20 20 |UNAVAILABLE
$0625/$XXXX :ERR|$0625: 20 20 20 |UNAVAILABLE
$0628/$XXXX :ERR|$0628: 20 20 20 |UNAVAILABLE
$062b/$XXXX :ERR|$062b: 20 20 20 |UNAVAILABLE
$062e/$XXXX :ERR|$062e: 20 20 12 |UNAVAILABLE
$0631/$XXXX :ERR|$0631: 05 01    |UNAVAILABLE
$0633/$0633 :ERR|$0633: 04       |$0633: 20 20 20
$0634/$XXXX :ERR|$0634: 19 2E 20 |UNAVAILABLE
$0637/$XXXX :ERR|$0637: 20 20 20 |UNAVAILABLE
$063a/$XXXX :ERR|$063a: 20 20 20 |UNAVAILABLE
$063d/$XXXX :ERR|$063d: 20 20 20 |UNAVAILABLE
$0640/$XXXX :ERR|$0640: 20 20 20 |UNAVAILABLE
$0643/$XXXX :ERR|$0643: 20 20 20 |UNAVAILABLE
```

```
$0646/$XXXX :ERR|$0646: 20 20 20 |UNAVAILABLE
$0649/$XXXX :ERR|$0649: 20 20 20 |UNAVAILABLE
$064c/$XXXX :ERR|$064c: 20 20 20 |UNAVAILABLE
$064f/$XXXX :ERR|$064f: 20 20 20 |UNAVAILABLE
$0652/$XXXX :ERR|$0652: 20 20 20 |UNAVAILABLE
$0655/$XXXX :ERR|$0655: 20 20 20 |UNAVAILABLE
$0658/$XXXX :ERR|$0658: 12       |UNAVAILABLE
$0659/$XXXX :ERR|$0659: 15 0E    |UNAVAILABLE
$065b/$XXXX :ERR|$065b: 20 20 20 |UNAVAILABLE
$065e/$XXXX :ERR|$065e: 20 20 20 |UNAVAILABLE
$0661/$XXXX :ERR|$0661: 20 20 20 |UNAVAILABLE
$0664/$XXXX :ERR|$0664: 20 20 20 |UNAVAILABLE
$0667/$XXXX :ERR|$0667: 20 20 20 |UNAVAILABLE
$066a/$XXXX :ERR|$066a: 20 20 20 |UNAVAILABLE
$066d/$XXXX :ERR|$066d: 20 20 20 |UNAVAILABLE
$0670/$XXXX :ERR|$0670: 20 20 20 |UNAVAILABLE
$0673/$XXXX :ERR|$0673: 20 20 20 |UNAVAILABLE
$0676/$XXXX :ERR|$0676: 20 20 20 |UNAVAILABLE
$0679/$XXXX :ERR|$0679: 20 20 20 |UNAVAILABLE
$067c/$XXXX :ERR|$067c: 20 20 20 |UNAVAILABLE
$067f/$XXXX :ERR|$067f: 20 20 20 |UNAVAILABLE
$0682/$XXXX :ERR|$0682: 20 20 20 |UNAVAILABLE
$0685/$XXXX :ERR|$0685: 20 20 20 |UNAVAILABLE
$0688/$XXXX :ERR|$0688: 20 20 20 |UNAVAILABLE
$068b/$XXXX :ERR|$068b: 20 20 20 |UNAVAILABLE
$068e/$XXXX :ERR|$068e: 20 20 20 |UNAVAILABLE
$0691/$XXXX :ERR|$0691: 20 20 20 |UNAVAILABLE
$0694/$XXXX :ERR|$0694: 20 20 20 |UNAVAILABLE
$0697/$XXXX :ERR|$0697: 20 20 20 |UNAVAILABLE
$069a/$XXXX :ERR|$069a: 20 20 20 |UNAVAILABLE
$069d/$XXXX :ERR|$069d: 20 20 20 |UNAVAILABLE
$06a0/$XXXX :ERR|$06a0: 20 20 20 |UNAVAILABLE
$06a3/$XXXX :ERR|$06a3: 20 20 20 |UNAVAILABLE
$06a6/$XXXX :ERR|$06a6: 20 20 20 |UNAVAILABLE
$06a9/$XXXX :ERR|$06a9: 20 20 20 |UNAVAILABLE
$06ac/$XXXX :ERR|$06ac: 20 20 20 |UNAVAILABLE
$06af/$XXXX :ERR|$06af: 20 20 20 |UNAVAILABLE
$06b2/$XXXX :ERR|$06b2: 20 20 20 |UNAVAILABLE
$06b5/$XXXX :ERR|$06b5: 20 20 20 |UNAVAILABLE
$06b8/$XXXX :ERR|$06b8: 20 20 20 |UNAVAILABLE
$06bb/$XXXX :ERR|$06bb: 20 20 20 |UNAVAILABLE
$06be/$XXXX :ERR|$06be: 20 20 20 |UNAVAILABLE
$06c1/$XXXX :ERR|$06c1: 20 20 20 |UNAVAILABLE
$06c4/$XXXX :ERR|$06c4: 20 20 20 |UNAVAILABLE
$06c7/$XXXX :ERR|$06c7: 20 20 20 |UNAVAILABLE
$06ca/$XXXX :ERR|$06ca: 20 20 20 |UNAVAILABLE
$06cd/$XXXX :ERR|$06cd: 20 20 20 |UNAVAILABLE
$06d0/$XXXX :ERR|$06d0: 20 20 20 |UNAVAILABLE
$06d3/$XXXX :ERR|$06d3: 20 20 20 |UNAVAILABLE
$06d6/$XXXX :ERR|$06d6: 20 20 20 |UNAVAILABLE
$06d9/$XXXX :ERR|$06d9: 20 20 20 |UNAVAILABLE
$06dc/$XXXX :ERR|$06dc: 20 20 20 |UNAVAILABLE
$06df/$XXXX :ERR|$06df: 20 20 20 |UNAVAILABLE
$06e2/$XXXX :ERR|$06e2: 20 20 20 |UNAVAILABLE
$06e5/$XXXX :ERR|$06e5: 20 20 20 |UNAVAILABLE
$06e8/$XXXX :ERR|$06e8: 20 20 20 |UNAVAILABLE
$06eb/$XXXX :ERR|$06eb: 20 20 20 |UNAVAILABLE
$06ee/$XXXX :ERR|$06ee: 20 20 20 |UNAVAILABLE
$06f1/$XXXX :ERR|$06f1: 20 20 20 |UNAVAILABLE
$06f4/$XXXX :ERR|$06f4: 20 20 20 |UNAVAILABLE
$06f7/$XXXX :ERR|$06f7: 20 20 20 |UNAVAILABLE
$06fa/$XXXX :ERR|$06fa: 20 20 20 |UNAVAILABLE
$06fd/$XXXX :ERR|$06fd: 20 20 20 |UNAVAILABLE
$0700/$XXXX :ERR|$0700: 20 20 20 |UNAVAILABLE
$0703/$XXXX :ERR|$0703: 20 20 20 |UNAVAILABLE
$0706/$XXXX :ERR|$0706: 20 20 20 |UNAVAILABLE
$0709/$XXXX :ERR|$0709: 20 20 20 |UNAVAILABLE
$070c/$XXXX :ERR|$070c: 20 20 20 |UNAVAILABLE
$070f/$XXXX :ERR|$070f: 20 20 20 |UNAVAILABLE
$0712/$XXXX :ERR|$0712: 20 20 20 |UNAVAILABLE
$0715/$XXXX :ERR|$0715: 20 20 20 |UNAVAILABLE
$0718/$XXXX :ERR|$0718: 20 20 20 |UNAVAILABLE
$071b/$XXXX :ERR|$071b: 20 20 20 |UNAVAILABLE
$071e/$XXXX :ERR|$071e: 20 20 54 |UNAVAILABLE
$cffa/$cffa :ERR|$cffa: 09 59    |$cffa: 0D 59 14
$cffc/$XXXX :ERR|$cffc: 14       |UNAVAILABLE
```

Dall'esame che ho effettuato ho potuto constatare che le differenze sono contenute essenzialmente in aree di memoria non collegate direttamente alla intro bensì, probabilmente, al processo di exomizzazione impiegato, ovvero in alcune locazioni in RAM da pagina 0 fino a pagina 7.

Qualche byte diverso lo rilevo anche nell'area agli indirizzi $CFFA-$CFFC che, dall'esame del codice sorgente, ho determinato essere deputata a contenere alcune variabili che ho adoperato nel mio progetto.

In conclusione, in questo specifico caso posso tranquillamente asserire che i due prg possono essere ricondotti allo stesso codice sorgente originario.

Sono consapevole che questo articolo è "tosto", è molto tecnico e quindi...non è per tutti.
Spero comunque di essere riuscito ad illustrare in maniera chiara ai lettori interessati a queste problematiche tutte le info necessarie ed i "ferri del mestiere" utili.

E' evidente che una analisi effettuata su procedure più complesse richiede più tempo ed attenzione. Inoltre, alcune tecniche, come ad esempio la sovrascrittura degli opcodes sulla memoria durante l'esecuzione del programma, non aiuterebbe coloro che volessero sfruttare questa metodologia di analisi.

Chi volesse approfondire scriva pure alla redazione di RetroMagazine World al seguente indirizzo email: retromagazine.redazione@gmail.com
Sarà mia cura rispondere, naturalmente.

Un caro saluto a tutti e alla prossima!

Fig. 4 - La intro che è stata oggetto della mia analisi, che è stata la prima intro che ho pubblicato su CSDB, "The Awekening of the Force", del 2016.

https://csdb.dk/release/?id=150842

Il gioco del "Mini-Lotto" (per C64)

di Eugenio Rapella

In questo articolo propongo una variazione di un programma che avevo presentato in un libro molti anni fa. È poco più di un pretesto per lavorare un po' con il nostro C64 sia in Basic sia in Assembly.
Le conclusioni a cui giungeremo tramite il nostro "Gioco del Lotto" semplificato rimangono valide per il gioco normale (liberandoci dalle complicazioni di carattere combinatorio).
Ecco la favoletta:

Molti cittadini di Bingotown giocano ogni settimana al Mini-Lotto. Dal lunedì al venerdì è possibile puntare su uno dei dieci numeri 0,1,2,..,9. Al sabato viene eseguita un'estrazione e coloro che hanno giocato il numero sorteggiato vincono. La somma per una giocata è fissa e piuttosto elevata; normalmente una persona non gioca più di un numero a settimana.
Tre amici, Alfio, Bruno, Carlo, sono fra i più incalliti giocatori: ogni settimana si recano al botteghino.

Alfio sostiene che la miglior strategia è quella di giocare sempre lo stesso numero perché, prima o poi, è destinato ad uscire. Poiché considera il quattro il suo "numero fortunato" (Alfio è nato il 4 aprile), gioca sempre e solo quello.

Bruno tiene aggiornata una "tabella dei ritardi" dove associa ad ogni numero il numero di settimane trascorse dall'ultima uscita. Quando si reca al botteghino gioca il numero con il ritardo maggiore: se un numero è un po' che non si fa vedere, ... beh, sta arrivando il suo turno!

Carlo sostiene che sono tutte baggianate e che giocare un numero o l'altro è del tutto indifferente. Con l'aiuto di un mazzo di carte ben mescolato, ogni settimana sceglie a caso il numero su cui puntare.

Che dire? Qual è la strategia più promettente? Ne parliamo un po' più avanti. Per ora mettiamo al lavoro il nostro C64: simulerà un bel po' di estrazioni e terrà aggiornata la situazione delle vincite dei tre amici.

Proporrò tre programmi: il primo (in Basic) mostra in dettaglio la situazione man mano che si succedono le estrazioni. È destinato a chi vuol seguire in tempo reale come stanno andando le cose. Con tutte le sue PRINT è, però, un bel po' ... lento.
Il secondo (sempre in Basic) è identico al primo "ripulito" dalle PRINT intermedie. Ci fornisce solo i risultati finali ma è, naturalmente, più veloce del primo.
Il terzo è identico al secondo, ma è in Assembly ed è... incredibilmente veloce. Non sono un esperto di Assembly per cui, molto probabilmente, il mio programma potrebbe essere migliorato, ma il mio obiettivo è, diciamo così, di tipo didattico. Un piccolo esperimento: un codice (ricco di commenti) che ho scritto basandomi quasi esclusivamente su un "Vocabulary of the 6510 Chip" che ho trovato in Internet.

Ecco qui il listato del primo programma:

```
100 poke 53280,6:poke 53281,6:print chr$(5):x=rnd(-ti):dim
r(9),f(9):pa=4
110 ra$=".. perso...": v$=chr$(18)+" vinto !! "+chr$(146)
115 for nl=1to39:s$=s$+chr$(32):next
120 print chr$(147):input"quante simulazioni ";ns:print chr$(147)
130 for k=1 to ns: m=0:x=int(rnd(1)*10):f(x)=f(x)+1:w$=""
140 print chr$(19):print"simul n.";k;"   su";ns
150 print"numero uscito:";x:print
160 print"-----------------------------------"
170 print"alfio ha puntato sul numero ";pa:rr$=ra$
180 if x=pa then rr$=v$:ta=ta+1
190 print"alfio ha";rr$:print"tot alfio";ta:print"% alfio";100*ta/k
200 print:print"-----------------------------------":m=0
210 print
220 print s$
230 for j=0 to 9:w$=w$+str$(r(j)):if r(j)>m then m=r(j):pb=j
240 next j:print chr$(145)+w$
250 print:print"bruno ha puntato sul numero ";pb:rr$=ra$
260 if x=pb then rr$=v$:tb=tb+1
270 print"bruno ha";rr$:print"tot bruno";tb:print"% bruno";100*tb/k
280 for j=0 to 9:r(j)=r(j)+1:next j
290 r(x)=0
300 pc=int(rnd(1)*10)
310 print:print"-----------------------------------"
320 print"carlo ha puntato sul numero ";pc:rr$=ra$
330 if x=pc then rr$=v$:tc=tc+1
340 print"carlo ha";rr$:print"tot carlo";tc:print"% carlo";100*tc/k
350 next k
360 print chr$(147):print"sono state eseguite ";ns;"simulazioni"
370 print:print"totale di alfio:";ta
380 print"% alfio:";100*ta/ns
390 print:print"totale di bruno:";tb
400 print"% bruno:";100*tb/ns
410 print:print"totale di carlo:";tc
420 print"% carlo:";100*tc/ns
430 print:for k=0 to 9:print"freq. di";k;">";f(k);" in %:";100*f(k)/
ns:next
```

Alla 100 si sceglie la combinazione di colori (bordo, sfondo, carattere), si inizializza il generatore di numeri casuali e si dimensionano i due vettori R e F, entrambi di dimensione 10. Per ogni I da 0 a 9, R(I) conterrà le settimane di ritardo del numero I (ad uso di Bruno) e F(I) la frequenza dell'uscita del numero I. PA è la variabile che contiene la puntata di Aldo che, sappiamo, è sempre e comunque il numero 4.
Alla 110 vengono definite un paio di costanti alfanumeriche che ci serviranno e, alla 120, viene richiesto il numero di simulazioni che si intendono eseguire (NS). Si ripulisce lo schermo e ... si parte!

Il gruppo di istruzioni dalla 130 (FOR K...) alla 350 (NEXT K) viene eseguito per ciascuna simulazione.
Alla 130, a X viene assegnato il numero sorteggiato e la "frequenza di X" viene subito aggiornata.
Ogni singola simulazione prevede la stampa di una intera schermata. Alla 140, la PRINT CHR$(19) riporta il cursore nell'angolo superiore sinistro dello schermo senza ripulirlo. Questo fa sì che, al variare delle simulazioni (circa una al secondo), sullo schermo si modifichino solo i valori che variano davvero. Le scritte che non devono cambiare rimangono fisse e consentono la lettura della schermata (se provate a sostituire CHR$(19) con CHR$(147) vedrete la differenza).
Viene poi stampato il numero della simulazione (K) e il numero uscito (X).

Si passa ora a definire le puntate dei tre amici e a verificare le eventuali vincite.

Per Alfio la cosa è immediata: se X=PA (PA=puntata di Alfio=4), il totale delle vincite di Alfio (TA) viene incrementato, la segnalazione ("ha vinto" / "ha perso") viene aggiornata; viene stampato TA e la percentuale di vittorie di Alfio fino a quel momento (100*TA/K).

Per scegliere cosa puntare, Bruno deve controllare qual è il numero con maggior ritardo. Di questo si occupano le istruzioni 230 e 240: al termine del ciclo FOR-NEXT, la variabile PB conterrà il numero (da 0 a 9) a cui è associato il massimo ritardo, la variabile M l'entità di tale ritardo (si passano in rassegna gli elementi di R(..): non appena R(J) supera l'attuale M, M e PB vengono aggiornati).
All'interno del ciclo FOR-NEXT viene anche costruita la stringa W$ che contiene la sequenza dei ritardi; W$ viene poi stampata (CHR$(145) spinge il cursore verso l'alto; in questo modo la stampa si sovrappone alla stringa della simulazione precedente sostituendola).
Dunque Bruno punta su PB; se X=PB (istr. 260) il totale delle vincite di Bruno viene incrementato. Le istruzioni 260-270 (relative a Bruno) hanno lo stesso effetto delle 180-190 (relative ad Alfio).
Alla 280 viene incrementato di una unità il ritardo di tutti i numeri tranne il ritardo del numero sorteggiato che, naturalmente, viene azzerato (istr. 290).

È ora il turno di Carlo. Sappiamo che Carlo sceglie a caso il numero da giocare per cui il C64 lo genera alla 300. Le istruzioni 310-340 sono analoghe a quelle relative agli altri due amici.
All'istruzione 360 si arriva solo quando tutte le simulazioni sono state eseguite. Per ciascuno degli amici viene stampato il numero delle giocate vincenti e le relative percentuali.
Infine vengono stampate le frequenze (assolute e percentuali) di sorteggio dei vari numeri.
In Fig.1 una schermata durante l'elaborazione (con la tabella dei ritardi utilizzata da Bruno); in Fig. 2 la schermata finale.

Le simulazioni confermano che nessuna delle strategie adottate dagli amici è preferibile alle altre. Poiché la probabilità di azzeccare la cifra esatta è del 10 %, anche le frequenze percentuali si avvicineranno a questo valore.
È vero che, se si gioca sempre lo stesso numero, o il numero con maggior ritardo, questo, prima o poi, uscirà, ma questo è semplicemente dovuto al fatto di ... giocare (qualsiasi numero): sbagliare sempre è poco probabile e, prima o poi, si vince.

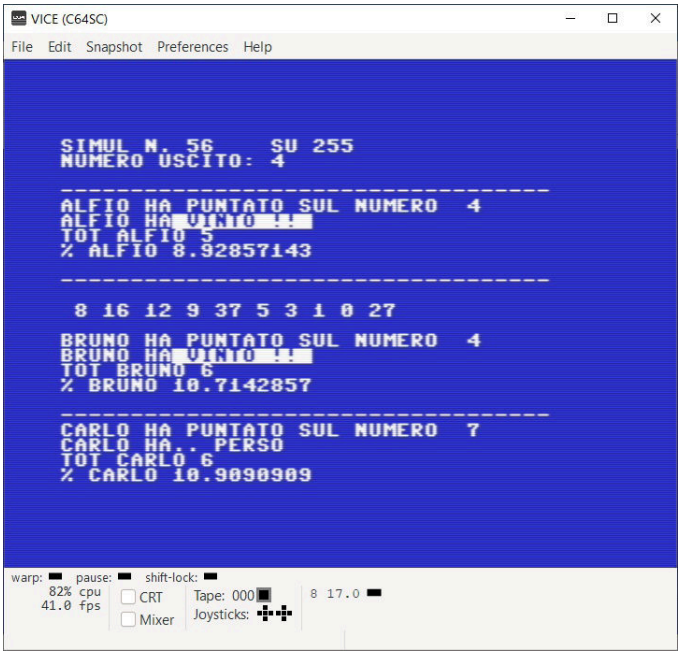


Fig. 1

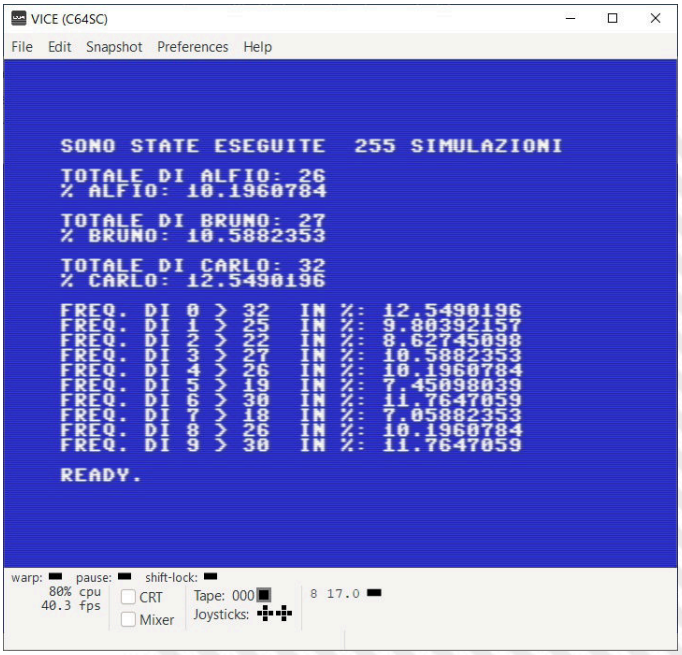


Fig. 2

Su diversi quotidiani vengono riportati i numeri estratti al “Gioco del Lotto”; un po’ di anni fa venivano segnalati anche i numeri con maggior ritardo. Solo recentemente, in edicola, ho scoperto l’esistenza di “Settimana Lotto”: ben 48 pagine (di gran formato) strapiene di statistiche, frequenze, ritardi, dati storici … di dubbia utilità! (In Fig. 3 e in Fig. 4 un paio di esempi).

Sappiamo che, quando lavora in Basic, il nostro Commodore non è un “Treno ad Alta Velocità”: per eseguire 255 simulazioni impiega circa cinque minuti e mezzo!

Se siamo interessati ai soli risultati finali, ecco qui sotto il programma “ripulito” che esegue 255 simulazioni in 1 minuto e 44 secondi (un risparmio di tempo del 68%).
Nella stesura di questi programmi ho utilizzato CBM Prg Studio (v. 4.2.1) con l’emulatore VICE (v. 3.8): i tempi che ho segnalato si riferiscono a questa situazione.

```
100 poke 53280,2:poke 53281,2:print chr$(5):x=rnd(-ti):dim
r(9),f(9):pa=4
110 print chr$(147):input"quante simulazioni ";ns:print"o.k.
procedo"
120 for k=1 to ns: m=0:x=int(rnd(1)*10):f(x)=f(x)+1
130 if x=pa then ta=ta+1
140 m=0
150 for j=0 to 9:if r(j)>m then m=r(j):pb=j
160 next j
170 if x=pb then tb=tb+1
180 for j=0 to 9:r(j)=r(j)+1:next j
190 r(x)=0
200 pc=int(rnd(1)*10)
210 if x=pc then tc=tc+1
220 next k
230 print chr$(147):print"sono state eseguite ";ns;"simulazioni"
240 print:print"totale di alfio:";ta
250 print"% alfio:";100*ta/ns
260 print:print"totale di bruno:";tb
270 print"% bruno:";100*tb/ns
280 print:print"totale di carlo:";tc
290 print"% carlo:";100*tc/ns
300 print:for k=0 to 9:print"freq. di";k;">";f(k);" in %:";
100*f(k)/ns:next
```

Ho provato a compilare questo programma con “Blitz Compiler”: la versione compilata esegue 255 simulazioni in 34 secondi.

Se però vogliamo prestazioni da “formula uno”, dobbiamo ricorrere all’Assembly.
Traduciamo in Assembly il contenuto del FOR-NEXT (istruzioni dalla 120 alla 220) del programma precedente; la stampa finale rimarrà in Basic.
Le locazioni di memoria a partire da $C000 possono ospitare il codice; per le variabili utilizzerò le locazioni $033C - $03FB (828-1019) che corrispondono al “buffer del registratore a cassetta” e, quindi, sono disponibili. Ho deciso di lasciare fisso il numero di simulazioni: 255; ecco come mi sono organizzato:

Fig. 3

Fig. 4

* il registro Y viene utilizzato esclusivamente come contatore delle simulazioni. Inizialmente viene caricato con il valore 255 (ldy #$ff) e viene decrementato (dey)

alla fine di ogni simulazione. Quando raggiunge il valore zero, il programma ha termine.

* Anche il registro X viene utilizzato, in più punti, come contatore: per azzerare alcune locazioni di memoria, per scorrere il "vettore delle frequenze", il "vettore dei ritardi" ecc.

* Le dieci locazioni dalla 828 ($033C) alla 837 ($345) conterranno le frequenze assolute di ciascuno dei numeri da 0 a 9. Se, in un dato istante, la locazione 828 contiene il numero 12, significa che, fino a quel momento, il numero 0 è stato sorteggiato 12 volte. Da notare che, essendo 255 simulazioni, qui, come in tutti gli altri casi, i numeri da memorizzare non superano questo valore e un byte è sufficiente a contenerli.

* Le dieci locazioni di memoria dalla 838 ($346) alla 847 ($34F) sono riservate al "vettore dei ritardi".

* Le locazioni 848 ($350), 849 ($351), 850 ($352) contengono rispettivamente TA, TB, TC: il "totale delle vincite di Alfio" ecc.

* La locazione 851 ($353) contiene il numero giocato da Bruno; la 852 ($354) contiene il valore massimo del "vettore dei ritardi".

* La locazione 853 ($355) è riservata al numero giocato da Carlo; la 854 ($356) contiene il numero estratto.

Per generare un numero casuale, ho utilizzato un codice Assembly che avevo trovato in Internet; sono dovuto ricorrere alla competenza di Marco Pistorio per svelarne i misteri.

Marco mi ha fornito una dettagliata spiegazione (che meriterebbe un articolo a sé stante) che cerco di sintetizzare.

Sappiamo che il nostro C64 è dotato del meraviglioso SID (Sound Interface Device) grazie al quale molti giochi sono accompagnati da fantastiche colonne sonore (andate a risentirvi quella di "Monty On The Run"). Il SID può anche generare un "rumore" ottenuto grazie ad un registro (LFSR = Linear Feedback Shift Register) il cui segnale, opportunamente manipolato, genera una sequenza pseudo-casuale di bit. Per far questo, occorre impostare la frequenza della "voice 3" nelle locazioni $D40E e $D40F:

```
lda #$ff; set voice #3 frequency
sta $d40e
sta $d40f
```

e impostare a 1 il bit #7 del registro che controlla la voce #3, portando a zero gli altri bit di questo registro:

```
lda #$80        ; set voice #3 control register
sta $d412       ; $80 = %10000000 (binary).
```

A questo punto, ogni volta che si andrà a leggere il contenuto della locazione $D41B (oscillator voice 3) si otterrà un valore pseudo-casuale nel range 0, 255.

Ecco il codice:

```
*=$c000             ; sys 49152
       lda #$ff     ; ********************
       sta $d40e    ; * inizializzazione
       sta $d40f    ; * del generatore di
       lda #$80     ; * di numeri casuali
       sta $d412    ; ********************

       ldx #$1b     ; le locazioni da
       lda #$0      ; 828 a 855 che
alfa1   sta $33c,x   ; verranno utilizzate
       dex          ; per le variabili
       bne alfa1    ; vengono preventivamente
       sta $33c     ; azzerate

       ldy #$ff     ; si eseguono 255 simulazioni
beta
       jsr teta     ; vai al sorteggio...
       sta $356     ; qui va il numero uscito
       tax          ; che va anche nel registro X
       inc $033c,x  ; aggiorna tot. vett. NUMERI USCITI
       cmp #$4      ; è uscito il 4?
       bne istr0
       inc $350     ; se sì, aggiorna il tot. di Alfio

; Nelle istruzioni che seguono viene determinato qual è
; il massimo valore nel VETTORE DEI RITARDI e a quale
; indice X corrisponde(sarà il numero che giocherà Bruno)
istr0   ldx #$09
gamma   lda $354     ; qui va l'attuale MASSIMO (M)
       cmp $346,x   ; confr con l'elemento del vett
RITARDI
       bcs istr1    ; se R(X)<=M vai a istr1
       lda $346,x   ; ... altrimenti
       sta $354     ; metti R(X) in M
       txa
       sta $353     ; e X in $353
istr1   dex
       bpl gamma    ; se non abbiamo finito si torna a
gamma

; Nelle istruzioni che seguono, il numero scelto da Bruno
; (quello con maggior ritardo), in $353, viene confrontato
; col numero uscito, in $356. Se sono uguali, il totale
delle
```

```
; vittorie di Bruno, in $351, viene incrementato.
       lda $353
       cmp $356
       bne istr2
       inc $351

; Nelle istruzioni che seguono viene aggiornato il
VETTORE DEI
; RITARDI: Si addiziona 1 a tutte le componenti, poi
viene
; azzerato il ritardo del numero uscito.
istr2   ldx #$09
istr3   inc $346,x
       dex
       bpl istr3
       ldx $356
       lda #$0
       sta $346,x

; Nelle istruzioni che seguono viene gestita la
situazione di
; Carlo: il numero che gioca è scelto a caso (jsr teta)
e, se
; coincide con il numero uscito, il suo totale (in $352)
viene
; incrementato.
       jsr teta
       cmp $356
       bne istr4
       inc $352

istr4   dey
       bne beta     ; se non hai finito, un'altra
simulazione.
       rts

;** subroutine teta: genera in A
;** una cifra casuale tra 0 e 9
teta
alfa    lda $d41b    ; in A un numero tra 0 e 255
       and #$0f     ; ora in A un numero tra 0 e 15
       cmp #$0a     ; è maggiore o uguale a 10 ?
       bcs alfa     ; se sì, torna ad alfa
       rts          ; in A c'è una cifra tra 0 e 9
```

Il programma in Assembly esegue le 255 simulazioni, ma non presenta i risultati ottenuti.
Uno dei "tools" di CBM Prg Studio è "Generate BASIC loader": il programma Assembly viene trasformato in una serie di DATA che il programma Basic generato leggerà e inserirà nelle locazioni a partire dallo SA (Starting Address) che corrisponde al *=$C000 del codice in Assembly. L'istruzione SYS SA manderà il tutto in esecuzione.
Ecco allora che possiamo completare il programma Basic generato con le istruzioni di stampa; anzi, possiamo fare di più.
Poiché le 255 simulazioni vengono eseguite molto velocemente, possiamo permetterci il lusso di far eseguire al C64 più "blocchi" di 255 simulazioni ciascuno e stampare i totali complessivi.

Ecco il codice:

```
1000 REM GENERATED ML LOADER
1040 poke 53280,4:poke 53281,4:print
chr$(5),chr$(147):dim f(9)
1120 SA = 49152
1160 FOR N = 0 TO 119
1200 READ A%:POKE SA+N,A%: NEXT N
1240 input"quanti gruppi di 255 simulazioni > ";g
1280 for g1=1 to g
1360 SYS SA
1400 ta=ta+peek(848)
1440 tb=tb+peek(849)
1480 tc=tc+peek(850)
1520 for k=0 to 9:f(k)=f(k)+peek(828+k):next
1560 next g1
1570 tgg=255*g
1600 print chr$(147):print"sono state
eseguite";tgg;"simulazioni"
1640 print:print"totale di alfio:";ta
1680 print"% alfio:";100*ta/tgg
1720 print:print"totale di bruno:";tb
1760 print"% bruno:";100*tb/tgg
1800 print:print"totale di carlo:";tc
1840 print"% carlo:";100*tc/tgg:print
1920 for k=0 to 9:print"freq. di";k;">";f(k);" in %:";
100*f(k)/tgg:next
1960 END
3060 DATA 169,255,141,14,212,141,15,212
3070 DATA 169,128,141,18,212,162,27,169
3080 DATA 0,157,60,3,202,208,250,141
3090 DATA 60,3,160,255,32,110,192,141
3100 DATA 86,3,170,254,60,3,201,4
3110 DATA 208,3,238,80,3,162,9,173
3120 DATA 84,3,221,70,3,176,10,189
3130 DATA 70,3,141,84,3,138,141,83
3140 DATA 3,202,16,235,173,83,3,205
3150 DATA 86,3,208,3,238,81,3,162
3160 DATA 9,254,70,3,202,16,250,174
3170 DATA 86,3,169,0,157,70,3,32
3180 DATA 110,192,205,86,3,208,3,238
3190 DATA 82,3,136,208,175,96,173,27
3200 DATA 212,41,15,201,10,176,247,96
```

Il FOR-NEXT (1160-1200) carica in memoria il codice

Assembly; alla 1240 si chiedono quanti “blocchi” si intendono eseguire. Per ogni blocco (FOR-NEXT, 1280-1560) viene eseguito il codice Assembly (SYS SA) e i totali delle vincite dei tre amici vengono aggiornati prelevando i valori dalle locazioni a loro riservati (1440-1480), lo stesso avviene per le frequenze assolute alla 1520.

Alla fine vengono stampati i risultati; il numero complessivo delle simulazioni eseguite è, naturalmente, pari a 255 per il numero di blocchi richiesti (TGG istr. 1570).

Se, dato il RUN, scegliete di eseguire un solo gruppo, eseguite 255 simulazioni e, come vedrete, il risultato è pressoché immediato (Fig. 5).

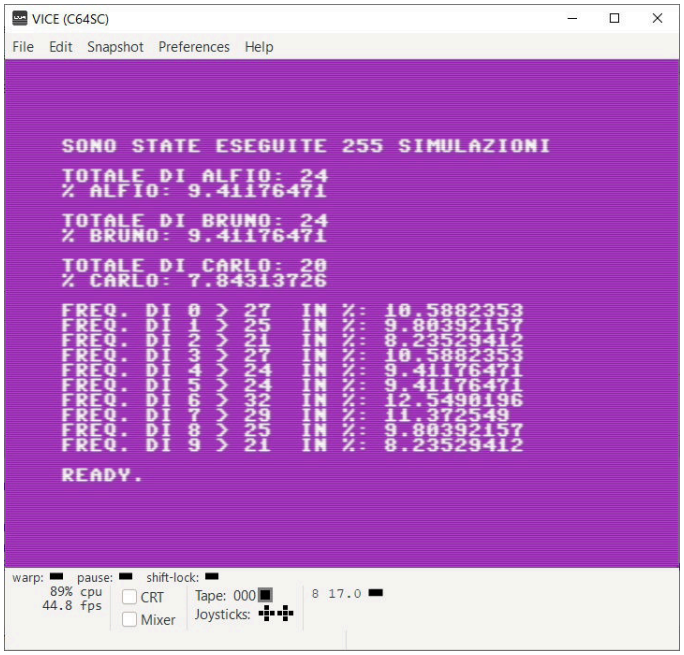


Fig. 5

Dunque, ricapitolando, per eseguire le 255 simulazioni, il primo programma ha impiegato 5 minuti e 30 secondi (ma c’erano davvero tante PRINT e lo scopo non era la velocità, ma mostrare come si evolveva la situazione); il secondo 1 minuto e 44 secondi; il secondo compilato 34 secondi e l’ultimo … 1 secondo!

Il programma in Assembly ci permette di … esagerare: se, al RUN, inseriamo la richiesta di 67 gruppi, operiamo ben 17085 giocate e, dopo circa 35 secondi, otteniamo (Fig. 6):

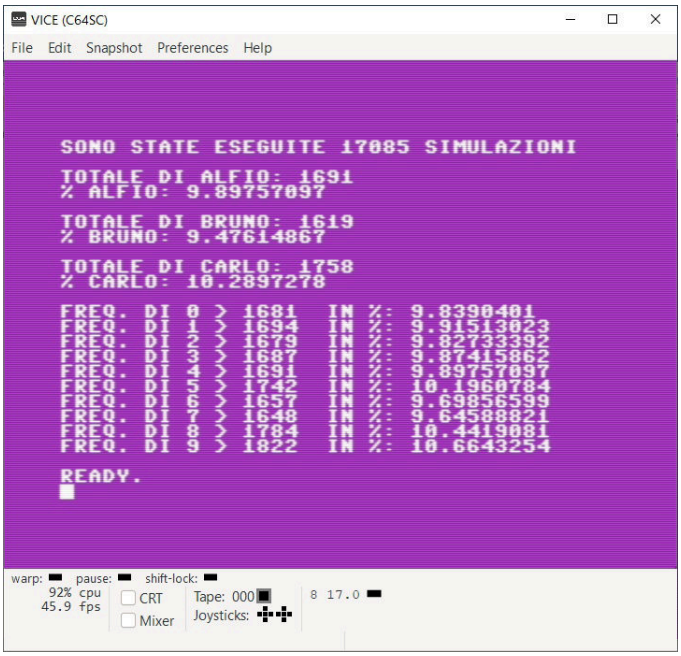


Fig. 6

… con l’aumentare del numero delle prove, le frequenze tendono ad assestarsi intorno al 10% a conferma, ancora una volta, che le strategie dei tre amici si equivalgono.

Eseguire 17085 simulazioni con gli altri programmi non è consigliato: quello “ripulito” compilato con Blitz Compiler richiederebbe circa 78 minuti, quello “ripulito” non compilato un’ora e 56 minuti e il primo programma … oltre sei ore!

Guida al Sega Genesis Development Kit (SGDK) - parte seconda

di Francesco Donatini

Nel precedente articolo avevamo preparato il setup e ottenuto la prima rom con la stampa a schermo del classico Hello World, adesso proseguiamo la nostra esplorazione del framework dedicato allo sviluppo di giochi per il Sega Megadrive con un altro classico del genere, la visualizzazione di uno sprite animato.

Fig. 1 - Tanta fatica per stordire uno scorpione... Non vi aspettavate mica Sonic?

Prima di iniziare richiamiamo alcune caratteristiche hardware inerenti gli sprites:

• le dimensioni minime di uno sprite corrispondono all'unità minimale che il chip video dedicato alla grafica (VDP) basato su caratteri (tiles) è in grado di gestire (8x8px), mentre le dimensioni massime corrispondono a 4x4 tiles (32x32px);

• per le risoluzioni standard che presentano 320px in orizzontale (320x224px in modalità NTSC, 320x240px in modalità PAL) possono essere visualizzati fino a un massimo di 80 sprites (non più di 20 per scanline);

• gli sprites sono visualizzati su un piano (plane/playfield) dedicato (quindi indipendente ad esempio dal background) di 512x512px, così da poter nascondere dalla visualizzazione di gioco gli sprite che non si vogliono visualizzare in quanto lo schermo di gioco inizia alle coordinate 128,128 ed essendo più piccolo occupa solo una porzione del piano degli sprites;

• all'interno della RAM dedicata alla parte grafica (VRAM) i dati degli sprite sono memorizzati in una tabella (sprite table) che vedremo a breve più nello specifico;

• uno sprite può essere associato ad una palette composta da sedici colori (di cui il primo trasparente) tra i 512 possibili (all'argomento palette dedicheremo un approfondimento in un successivo articolo).

PREPARIAMO LO SCORPIONE

Come sprite di esempio utilizzeremo quello di un simpatico scorpione preso da un gioco che sto realizzando, per il quale sono state previste le animazioni per il movimento e per quando è in stato di stun. Quando preparate il file PNG è indispensabile configurarne i colori con la modalità di codifica indicizzata (indexed color mode) e utilizzare una palette composta al massimo da 16 colori, in questo caso è stato utilizzato il viola che è il colore adottato in genere come trasparente per convenzione (valori RGB 255,0,255).

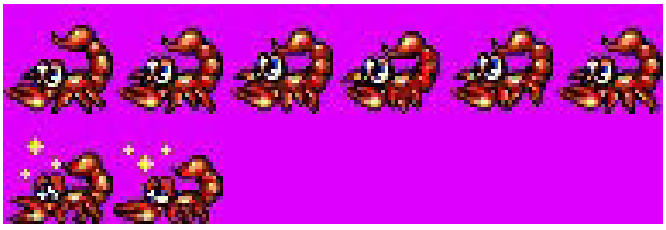


Fig. 2 - Lo spritesheet con le animazioni

Una volta preparata l'immagine, nella cartella "res" creiamo il file "resources.res" che è identificato dal SGDK come il file di configurazione delle risorse impiegate nel gioco (immagini, audio, ecc.) e inseriamo il riferimento alla nostra immagine dello scorpione definendolo con questi attributi:

1. SPRITE è la tipologia di risorsa che sarà appunto uno sprite (fosse stata un'immagine statica avremmo utilizzato IMAGE);
2. spr_scorpion è il nome della costante di tipo

"SpriteDefinition" che avremo a disposizione nel codice per gestire lo sprite;

3. "sprites/scorpion.png" è il path del file PNG relativo alla posizione della cartella "res";

4. i due numeri "4" indicano il numero di tiles orizzontali e verticali di cui è composto un singolo frame dello sprite, in questo caso 4 e 4 essendo le dimensioni del singolo frame 32x32px.

La riga di configurazione della risorsa di tipo Sprite:

```
SPRITE spr_scorpion "sprites/scorpion.png" 4 4
```

Ci sono ulteriori parametri che possono essere specificati che non sono di nostro interesse in questo momento. Importante è invece sapere che in fase di compilazione il file di configurazione viene processato da un programma Java che opererà una serie di ottimizzazioni sulle risorse multimediali fornite trasformandole in formato assembly direttamente leggibile dal framework. Trovate tutti i dettagli nel file di testo "rescomp.txt" incluso nel framework, di cui consiglio caldamente la lettura.

VISUALIZZIAMO LO SCORPIONE

Per visualizzare a schermo lo scorpione possiamo riprendere il codice dello scorso articolo, eliminare il comando di stampa della frase e (trovate il codice in Fig.4):

• innanzitutto compiliamo, se tutto è stato realizzato in maniera correttà sarà generato nella cartella "res" un file "resources.h" in cui troviamo definita come extern la nostra costante di tipo "SpriteDefinition" dello scorpione;

• (A) aggiungiamo l'include dell'header del file delle risorse generato così da renderlo disponibile al programma;

• (B) Inizializziamo il VDP e impostiamo lo schermo ad esempio con risoluzione NTSC standard 320x224px;

• (C) inizializziamo lo sprite engine del framework;

• (D) attribuiamo una palette (in questo caso la prima che è PAL0) con la palette presa direttamente dall'immagine dello scorpione; se provate a caricare il programma prima e dopo aver inserito questa istruzione, vedrete che il colore di sfondo da nero diventerà viola, questo perché di default il colore di sfondo dello schermo è il primo della prima palette;

```c
#include <genesis.h>
// (A)
#include <resources.h>

int main()
{
    // (B)
    VDP_init();
    VDP_setScreenWidth320();
    VDP_setScreenHeight224();

    // (C)
    SPR_init();

    // (D)
    PAL_setPalette(PAL0, spr_scorpion.palette->data, DMA);

    // (E)
    Sprite* scorpionSprite = SPR_addSpriteSafe(&spr_scorpion, 100, 100, TILE_ATTR(PAL0, FALSE, FALSE, FALSE));

    while (TRUE)
    {
        // (F)
        SPR_setAnimAndFrame(scorpionSprite, 0, 0);

        // (G)
        SPR_update();

        SYS_doVBlankProcess();
    }

}
```

Fig. 4 - Il codice per visualizzare lo scorpione a schermo

• (E) Inizializziamo il nostro sprite utilizzando la definizione proveniente dalle risorse, piazzandolo in posizione 100,100 (la posizione si riferisce al punto all'estremo in alto a sinistra del frame) e attribuendogli la PAL0 (il resto dei parametri riguarda la priorità e il flip che per il momento tralasciamo);

• (F) impostiamo l'animazione dello scorpione, ad esempio la prima quindi quella con indice zero, e il frame di partenza, ad esempio il primo quindi anche qui quello con indice zero;

• (G) chiamiamo l'update dello sprite engine così che i dati della sprite table possano essere trasferiti al VDP all'occorrere del vertical interrupt.

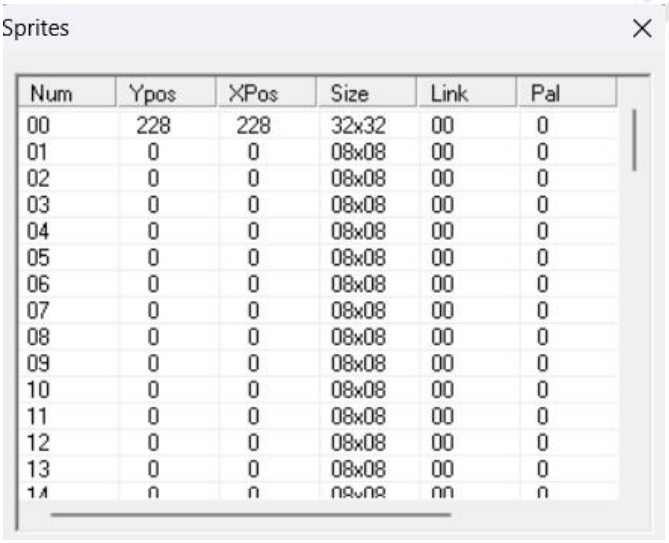


Fig. 5 - La tabella degli sprites dell'emulatore Gens_KMod

Come risultato abbiamo ottenuto il nostro scorpione che è bello fermo immobile ma comunque presente e correttamente visualizzato all'interno dello schermo di gioco.

UNO SGUARDO ALLA SPRITE TABLE

Se eseguiamo il programma utilizzando l'emulatore Gens_KMod (che avevo segnalato nel precedente articolo come strumento di debug) possiamo visualizzare la tabella degli sprites accedendo da CPU → Debug → Genesis → VDP - Sprites, ciascuna entry è costituita da 8 bytes in cui sono rappresentati:

• "Num" è il numero dello sprite, il nostro essendo il primo ha il numero 0, e come già detto essendo la risoluzione di 320px in orizzontale abbiamo a disposizione fino ad un massimo di 80 sprites;

• "YPos" e "Xpos" sono la posizione verticale ed orizzontale dello sprite, il valore è 228 perché il piano dedicato agli sprites inizia dalla posizione 128,128 rispetto allo schermo per cui la posizione sullo schermo 100,100 corriponde alla 228,228 del piano degli sprites;

• "Size" è la dimensione dello sprite quindi 32x32px;

• "Link" è una internalità che indica collegamento al successivo sprite della linked list della tabella;

• "Pal" è la palette dello sprite e quindi il valore è 0;

• "Tile" è il numero iniziale della tile usata dallo sprite;

• "Flags" sono dei valori 0/1 relativi alla priorità alta/bassa e all'eventuale flip orizzontale/verticale;

Vedrete che la lettura dei dati della tabella degli sprites diventerà sempre più una costante man mano che i vostri giochi si arrichiranno di elementi grafici e di interazioni, torneremo sull'argomento con qualche esempio preso da situazioni reali.

ANIMIAMO LO SCORPIONE

È giunto il momento di riprendere lo scorpione che avevamo abbandonato ed animarlo stordendolo, poverino, con l'animazione di stun (trovate il codice in Fig. 6):

• (A) introduciamo due variabili, "frame" per memorizzare il frame corrente da visualizzare e "nextFrameTimer" per regolare la velocità del cambio frame (altrimenti l'animazione sarebbe troppo veloce);

• (B) nel loop di gioco, stabiliamo arbitrariamente a 10 il valore che il timer deve raggiungere per far scattare il frame successivo di animazione, e regoliamo il valore del frame in modo che non sia mai diverso da 0 e 1 in quanto l'animazione di stun si compone di due soli frames di animazione;

• (C) impostiamo l'animazione con lo stun e il frame corrente e avanziamo il timer del cambio frame;

• (D) l'update dello sprite engine viene spostato nel loop di gioco in quanto la tabella degli sprite deve recepire i cambiamenti dello sprite animato.

Questo è un semplice esempio di come è possibile realizzare una animazione di velocità costante, poi esaminando come sono fatte le struct Sprite e SpriteDefinition del framework potete costruirvi il vostro engine per, ad

```c
int main()
{
    // (A)
    s16 frame = 0;
    u8 nextFrameTimer = 0;

    VDP_init();
    VDP_setScreenWidth320();
    VDP_setScreenHeight224();

    SPR_init();

    PAL_setPalette(PAL0, spr_scorpion.palette->data, DMA);

    Sprite* scorpionSprite = SPR_addSpriteSafe(&spr_scorpion, 100, 100, TILE_ATTR(PAL0, FALSE, FALSE, FALSE));

    while (TRUE)
    {
        // (B)
        if (nextFrameTimer >= 10) {
            frame++;
            nextFrameTimer = 0;
        }
        if (frame == 2) {
            frame = 0;
        }

        // (C)
        SPR_setAnimAndFrame(scorpionSprite, 1, frame);

        nextFrameTimer++;

        // (D)
        SPR_update();

        SYS_doVBlankProcess();
    }

}
```

Fig. 6 - Il codice per animare lo scorpione

esempio, muovere lo sprite, stabilire velocità e direzione delle sequenze di animazione e realizzare tante altre mirabilie: in generale, il framework mette a disposizione i mattoncini senza offrire la “pappa scodellata” per cui tocca noi poi l’onere e l’onore di costruire la casa come vogliamo.

RISORSE UTILI

https://github.com/Stephane-D/SGDK

https://plutiedev.com/

https://danibus.wordpress.com/

https://www.copetti.org/writings/consoles/mega-drive-genesis/

https://segaretro.org/Sega_Mega_Drive/Technical_specifications

PunyInform: installazione degli strumenti e compilazione di un gioco

di Fredrik Ramsberg - adattato per RetroMagazine World da Gianluca Girelli

Nel tutorial precedente ho fornito una breve introduzione a PunyInform, una nuova libreria per scrivere avventure testuali per computer a 8 bit (e altri). Ora diamo un'occhiata a cosa devi scaricare e installare per iniziare a scrivere un gioco utilizzando PunyInform.

Per questo articolo utilizzerò Windows, ma puoi anche utilizzare PunyInform su Mac o Linux.

Passaggio 1: installa la libreria PunyInform

Vai su https://github.com/johanberntsson/PunyInform/releases e scarica la versione più recente di PunyInform (la versione che arriva per prima nella pagina) facendo clic su "Codice sorgente (zip)" in Risorse. Una volta terminato il download, fai clic sull'archivio Zip nella finestra del browser per aprire l'archivio in Esplora risorse, fai clic su "Estrai tutto" nella parte superiore della finestra, appena sotto la barra dei menu. Scegli una cartella in cui desideri posizionare i file PunyInform. Esplora risorse probabilmente suggerirà qualcosa come "C:\Users\MyUsername\Downloads\PunyInform-1_2\" e dovrebbe funzionare correttamente. Puoi sempre rinominare e/o spostare la cartella in un secondo momento.

Passaggio 2: ottieni il compilatore

PunyInform è una libreria per il linguaggio di programmazione Inform 6. Per utilizzarlo sarà necessario scaricare il compilatore, chiamato “inform” o “inform6”. PunyInform richiede almeno la versione 6.34 del compilatore. Se usi Windows, è disponibile come eseguibile binario, quindi puoi semplicemente scaricarlo e iniziare a usarlo. Vai a http://www.ifarchive.org/indexes/ifarchiveXinfocomXcompilersXinform6Xexecutables.html e scarica inform634_win32.zip . Apri l'archivio Zip ed estrai il file (ce n'è solo uno) nella cartella base di PunyInform (la cartella che contiene le sottocartelle “documentation” e “lib”).

Ecco come potrebbe apparire ora la tua cartella PunyInform [Fig. 1]:

Passaggio 3: installare un interprete Z-code

Anche se il vero punto di forza di PunyInform è che ti permette di scrivere giochi abbastanza veloci e abbastanza piccoli da poter essere giocati su 8 bit, i giochi prodotti saranno ugualmente riproducibili praticamente su qualsiasi

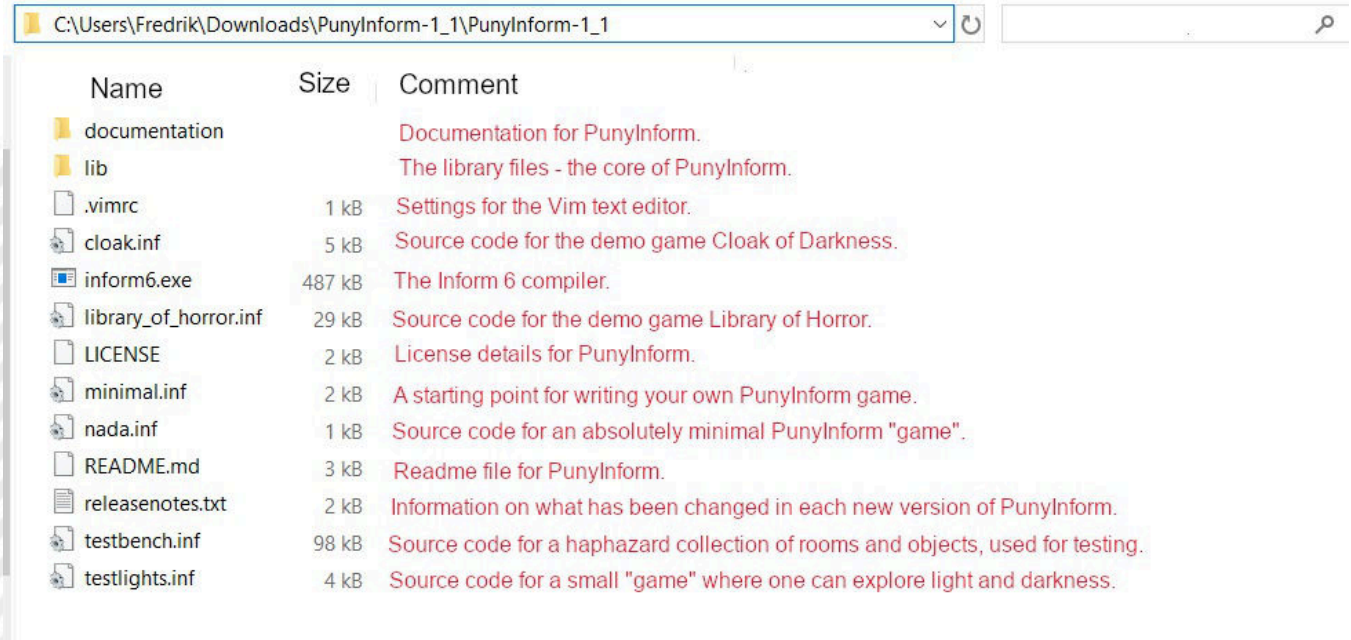
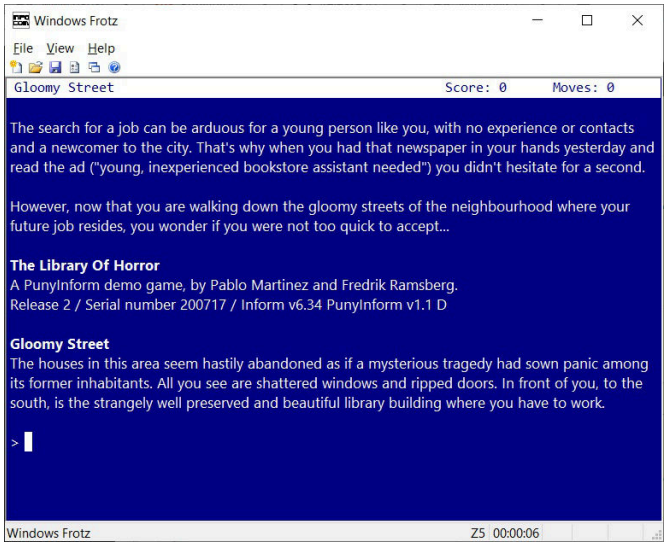


Fig. 1 - La cartella PunyInform

computer, e presto scoprirai di poter eseguire la maggior parte dei test sul tuo PC moderno. Per fare questo avrai bisogno di un interprete Z-code. PunyInform compila il codice sorgente di Inform 6 in Z-code, un formato di gioco inventato dai maghi delle avventure testuali di Infocom. Un file Z-code è infatti un'immagine di memoria per un computer virtuale/immaginario chiamato Z-machine. Sebbene la Z-machine non esista, può essere emulata su un'ampia gamma di piattaforme, inclusi i computer a 8 bit degli anni Ottanta, ma anche computer a 16 bit come Commodore Amiga e Atari ST, nonché computer con sistema operativo MS-DOS, qualsiasi versione di Windows, Mac OS, Unix, Linux, Android: qualunque sia la tua scelta. Ci vuole solo un interprete Z-code (un emulatore per Z-machine) per quella piattaforma. Esiste un interprete molto popolare chiamato Frotz, che è stato portato su molte piattaforme diverse. Per Windows, ti consiglio di procurarti Windows Frotz.



Vai su http://www.davidkinder.co.uk/frotz.html, scarica Windows Frotz ed esegui il programma di installazione. Quindi avvia Windows Frotz una volta, per assicurarti che abbia impostato correttamente le associazioni dei tipi di file.

Passaggio 4: inizia a scrivere il tuo primo gioco

In Esplora risorse, crea una copia del file minimal.inf e chiamalo mygame.inf . Quindi apri mygame.inf nel Blocco note (o l'editor di testo che preferisci – se vuoi qualcosa di veramente carino, puoi ottenere l'editor di testo Atom e installare il pacchetto "language-inform6" in File -> Impostazioni). Il file mygame.inf ora dovrebbe assomigliare a questo [Fig. 3].

Ora cambia il testo nella riga che inizia con Titolo costante (riga 9 dall'alto). Cambia Un gioco di esempio in Questo è il mio primo gioco o qualcos'altro che trovi divertente. Salvare il file.

Passaggio 5: compila il gioco

Avvia un prompt dei comandi (in Windows 10, fai clic sull'icona Start di Windows sulla barra dei menu di Windows e digita cmd, quindi fai clic su Prompt dei comandi.) Ora usa il CD dei comandi per andare alla cartella base di PunyInform. Se si trova su un dispositivo diverso, puoi digitare la lettera del dispositivo e ":" per passare prima a quel dispositivo, come D: e poi CD per andare alla cartella giusta. CD .. sale di un livello e CD \ (che è un carattere barra rovesciata) va alla cartella principale del dispositivo corrente. Se inizi a scrivere il nome di una cartella, puoi premere più volte il tasto TAB per visualizzare tutti i file e le cartelle con un nome corrispondente. Utilizzare DIR per elencare i file nella cartella corrente [Fig. 4].

Ora, per compilare il codice sorgente, digita:

```
inform6 -v3 +lib mygame
```

Questo dice al compilatore Inform 6 di leggere il codice sorgente in mygame.inf e compilarlo in un file Z-code versione 3 (un cosiddetto "file storia") chiamato mygame.z3, e di cercare i file di libreria nella cartella lib. Sono disponibili molte altre opzioni della riga di comando per il compilatore Inform 6. Per elencarli tutti, digita inform6 -h2 . La maggior parte delle volte, quando utilizzo PunyInform, mi ritrovo

```
12    ! $ZCODE_LESS_DICT_DATA=1 removes an empty data byte for every dictionary word.
13
14    Constant Story      "Minimal";
15    Constant Headline   "^A sample game which uses PunyInform.^";
16
17    ! Uncomment ONE of the two following lines, to show either time or score/turns
```

Fig. 3 - Il file mygame.inf

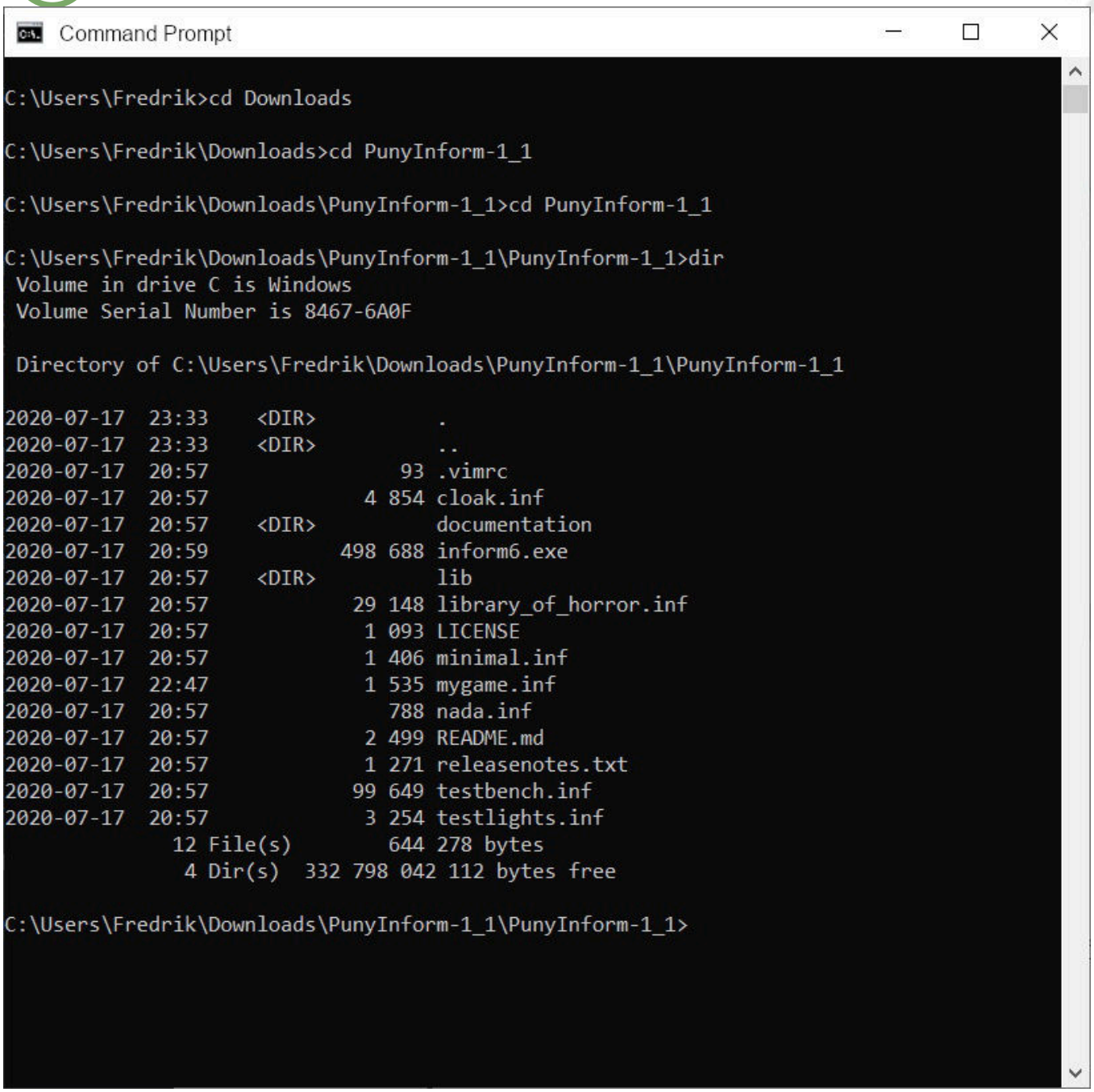


Fig. 4 - I file nella cartella PunyInform

a utilizzare le opzioni della riga di comando -e (per utilizzare le abbreviazioni delle stringhe, rendendo il file della storia più piccolo) e -s (per stampare alcune statistiche sul file della storia prodotto).

Passaggio 6: avvia il gioco in Windows Frotz

Ora usa Esplora risorse per andare alla cartella base PunyInform. Dovresti vedere mygame.z3 lì. Fare doppio clic su di esso. Il gioco ora dovrebbe avviarsi in Windows Frotz [Fig. 5].

Passaggio 7: trasforma il tuo gioco in un'immagine disco C64

Ora creiamo una versione Commodore 64 del tuo gioco. Per fare ciò, raggrupperemo il file della storia con un interprete Z-code chiamato Ozmoo. Vai a

https://ozmoo.online/

Nella sezione "Carica file Z-code", seleziona il file mygame.z3 e fai clic su "Carica". Quindi, in Costruisci gioco, assicurati che il file della storia mygame.z3 sia selezionato, quindi scorri fino alla fine della pagina e fai clic su "Costruisci". Se tutto va bene, riceverai un messaggio che il gioco è stato creato con successo e sotto File puoi vedere mygame.d64. Fare clic sul nome del file per scaricare il file.

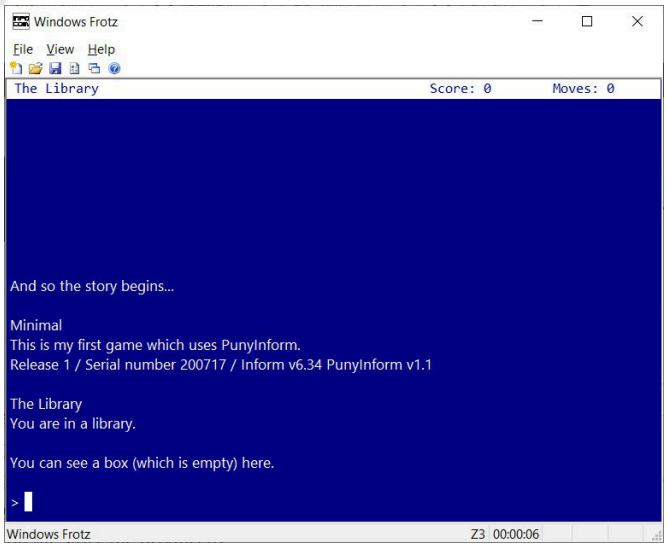


Fig. 5 - Il gioco in esecuzione su Windows Frotz

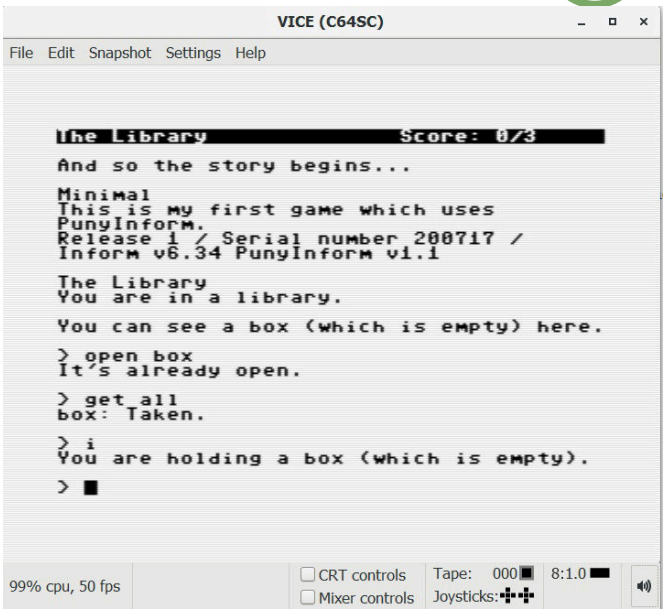


Fig. 6 - Lo stesso gioco in esecuzione sul Commodore 64

Passaggio 8: prova il tuo gioco su un C64 (emulato)

Se non hai già installato un emulatore Commodore 64, questo dovrebbe aiutarti. Scarica WinVICE da: https://viceemu.sourceforge.io/index.html#download e decomprimilo in una cartella a tua scelta. Potrebbe essere necessario installare 7-Zip per decomprimere WinVICE.

Quindi vai a questa cartella in Esplora risorse e fai doppio clic su x64sc.exe . Fai clic su File -> Allega immagine disco -> Unità n. 8 e punta a mygame.d64. Quindi digitare load"*",8 e premere Invio. Se ti stanchi del tempo di caricamento (realistico), puoi premere Alt-W per attivare la modalità warp (velocità massima).

Una volta terminato il caricamento del gioco, assicurati che la modalità Warp sia disattivata, digita esegui e premi Invio.
Il gioco ora si decomprime e poi si avvia [Fig. 6].

Risorse

Main project page per PunyInform:
https://github.com/johanberntsson/PunyInformPunyInform

Releases:
https://github.com/johanberntsson/PunyInform/releasesPunyInform

Manual:
https://github.com/johanberntsson/PunyInform/wiki/ManualInform

Designer's Manual, 4^ edizione:
https://www.inform-fiction.org/manual/html/

Un forum amichevole dove chiedere aiuto:
https://intfiction.org/

Un server Discord per supporto e discussioni su PunyInform:
https://discord.gg/zyYvb8JjE6Ozmoo

Un interprete Z-code utilizzabile per rendere giochi PunyInform fruibili su Commodore64/128/Plus4 e MEGA65:
https://ozmoo.online/

Pagina ufficiale PunyJam#4:
https://itch.io/jam/punyjam-4

Il Fortran sul Commodore 64 - il comando MEM

di Francesco Fiorentini

Continuano i nostri esperimenti con il Fortran utilizzando il compilatore **Abacus Fortran-64** sul Commodore 64.
Nei numeri precedenti abbiamo visto come scrivere, compilare ed eseguire programmi di crescente complessitá con questo strumento; questa volta peró proveremo a spingerci un po' piú in lá.
Chi programma in Basic sul Commodore 64, sa come questo linguaggio sia estremamente limitato su questa macchina e pertanto occorra ricorrere a due comandi particolari per leggere e scrivere direttamente in memoria sbloccando alcune funzioni altrimenti inutilizzabili.
Ovviamente mi riferisco ai comandi **PEEK** (per leggere) e **POKE** (per scrivere).
L'Abacus Fortran-64 si comporta esattamente alla stessa maniera: il set di comandi base é limitato, ma fortunatamente ci é stato dato modo di accedere direttamente alla memoria tramite il comando MEM che emula esattamente quello che fanno PEEK e POKE in Basic.

Il comando MEM

Il manuale dell'Abacus Fortran-64 ci descrive il comando MEM in questo modo:

SCOPO: esaminare o modificare singole locazioni di memoria

```
MEM (varnam|costante) = espressione intera
varnam = MEM (varnam |costante)
```

dove: l'argomento tra le () è un indirizzo compreso tra 1 e 65535 e varnam è una variabile intera

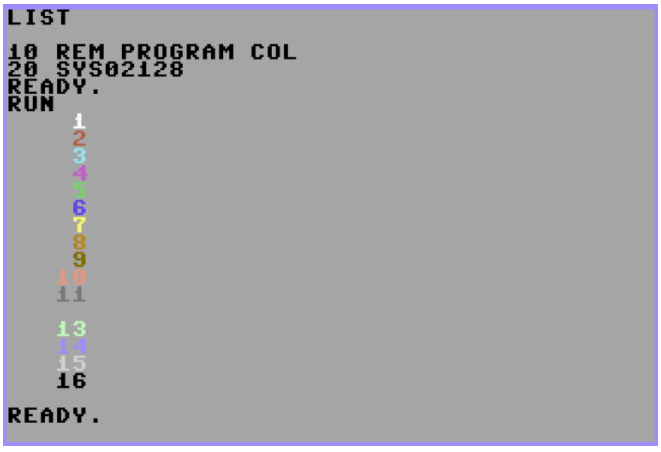
NOTE:
1) L'uso di MEM sul lato sinistro di un'equazione è equivalente al POKE in BASIC. L'uso di MEM sul lato destro di un'equazione equivale al PEEK in BASIC.

2) Le posizioni di memoria possono contenere solo valori compresi tra 0 e 255. Solo il byte inferiore dell'intero viene trasferito in memoria. Ottenendo un valore da MEM si memorizza uno 0 nel byte superiore della variabile intera.

Vediamo subito un esempio pratico per utilizzare il comando MEM nei nostri programmi Fortran:

```
10 : program col
15 :* this program writes
16 :* a list of numbers
17 :* from 1 to 16
18 :* with different colors
20 : integer a,b
30 : open 3,3
40 : do 10 i = 1, 16, 1
45 : mem (646) = i
50 : write (3,100) i
60 : 10 continue
65 : close 3
70 : stop
80 : 100 format(i5)
90 : end
```

Questo semplice programma utilizza il comando MEM per alterare il valore dell'indiirizzo di memoria 646.
L'indirizzo 646 (esadecimale: $0286) contiene il codice colore per il colore corrente del cursore: il Kernal applica il codice colore qui memorizzato alla RAM del colore del testo per ogni carattere visibile stampato sullo schermo.



La locazione 53270

Tratto dal libro "Peek Poke Commodore 64 manuale" di Hannes Rügheimer e Christian Spanik - speciale CHIP N° 9 Ottobre 1985.

In questo registro si trovano molte funzioni importanti : lo scroll fine orizzontale, che in linea di massima funziona come quello verticale, la selezione del MULTICOLOR, in cui ben 4 colori possono essere usati per ogni punto dello schermo.

I bit da 0 a 2 definiscono il movimento orizzontale del video. Il valore che tali 3 bits possono codificare va dal minimo 0 al massimo 7.

Il valore letto normalmente e' 0.

Per modificare tali bit potete scrivere:

```
POKE 53270,(PEEK<53270)AND243)0R(num.)
```

L'esempio qui sotto, in Basic V2, prende spunto da un piccolo programma pubblicato nel libro. Utilizzando lo scroll fine orizzontale possiamo simulare un terremto a video.

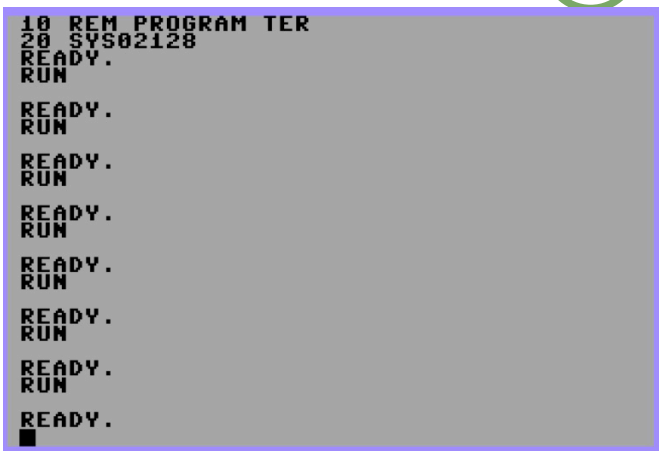
```
5 for i =1 to 50
10 x=x+1: if x=7 then x=-x
20 poke53270,(peek(53270) and 240) or abs(x)
30 next i
```

Questo é un ottimo esempio di come utilizzare il comando PEEK e POKE. Se volessimo replicare il programma in Abacus Fortran-64 dovremmo scrivere:

```
10 : program ter
15 :* this program simulates
17 :* an earthquake on the screen
20 : integer x,y
40 : do 10 i = 1, 50, 1
45 : x=x+1
46 : if (x .eq. 7) then
47 : x=-x
48 : endif
50 : y=(mem (53270) .and. 240) .or. abs(x)
55 : mem (53270)=y
60 : 10 continue
70 : stop
90 : end
```

In questo programma abbiamo introdotto alcuni nuovi comandi Fortran.

Il costrutto **IF (espressione logica) THEN ... ENDIF**, che ha una connotazione piú efficace di quella del corrispettivo in Basic, in quanto permette di utilizzare molteplici righe all'interno.



Output del programma TER in Abacus Fortran-64. Si noti la distanza del testo dal lato sinistro dello schermo.

Espressione logica puó invece utilizzare i seguenti operatori:

.EQ. - equal

.NE. - not equal

.LT. - less than

.GT. - greater than

.GE. - greater than or equal

La funzione ABS(), che corrisponde a quella del Basic.

Ed infine gli operandi AND e OR, che in Abacus Fortran-64 si scrivono rispettivamente **.AND.** e **.OR.**

Il programma Helic

La domanda a questo punto sorge spontanea... Possiamo utilizzare il comando MEM per disegnare ed animare sprite in Fortran? La risposta é ovviamente affermativa.

Il programma qui sotto dimostra come sia possibile disegnare uno sprite, in questo caso specifico un elicottero, e muoverlo per lo schermo utilizzando il joystick in porta 2. L'esempio e' tratto da un programma allegato al disco dell'Abacus Fortran-64.

```
1000 : *program sprite display
1010 : *use joystick in port 2
1020 : *to move sprite
1030 :     program helic
1040 :     character a
1050 :     dimension ia(189)
1060 : *sprite data facing left
1070 :     data ia/189*0/
1080 :     data ia/0,255,255,0,0,128,96/
1090 :     data ia(8)/0,128,144,1,240/
1100 :     data ia(13)/159,255,200,103,255/
1110 :     data ia(18)/254,0,14,127,0/
1120 :     data ia(23)/6,127,0,3,254,0/
```

```
1130 :     data ia(29)/0,32,0,15,255/
1131 : *sprite data facing right
1132 :     data ia(65)/255,255,0,1,0,0,1/
1133 :     data ia(72)/0,6,15,128,9,19/
1134 :     data ia(78)/255,249,127,255/
1135 :     data ia(82)/230,254,112,0,254/
1136 :     data ia(87)/96,0,127,192,0,4/
1137 :     data ia(93)/0,0,255,240,0/
1140 : *clear the screen
1150 :     do 15 j=1,1000
1160 :       i=1023+j
1170 :       mem(i)=32
1180 : 15  continue
1190 : *enter sprite data into memory
1200 :     do 10 j=1,189
1210 :       i=12287+j
1220 :       mem(i)=ia(j)
1230 : 10  continue
1240 : *define sprite location
1250 :     mem(2040)=192
1260 : *define the color
1270 :     mem(53287)=3
1280 : *define x-position
1290 :     mem(53248)=100
1300 : *define y-position
1310 :     mem(53249)=100
1320 : *reset x-position msb
1330 :     mem(53264)=0
1340 : *enable sprite
1350 :     mem(53269)=1
1360 : *read joystick
1370 : 20  ib=mem(56320)
1380 :     if (ib-127) 30,20
1390 : 30  if (ib-125) 40,50,60
1400 : 40  if (ib-119) 70,80,100
1410 :     goto 20
1420 : *down (y-pos)
1430 : 50  id=mem(53249)
1440 :     if (id-250) 55,20
1450 : 55  mem(53249)=id+1
1460 :     goto20
1470 : *up (y-pos)
1480 : 60  id=mem(53249)
1490 :     if (id-0) 20,20,65
1500 : 65  mem(53249)=id-1
1510 :     goto20
1520 : *button=quit
1530 : 70  mem(53269)=0
```

Output del programma HELIC

```
1540 :     stop
1550 : *right (x-pos)
1555 : 80  mem(2040)=194
1557 :     mem(2040)=192
1560 :     ic=mem(53248)
1570 :     ie=mem(53264)
1580 :     if (ie-0) 82,82,83
1590 : 82  if (ic-255) 90,85
1600 : 83  if (ic-89) 90,20
1610 : 85  mem(53264)=1
1620 :     mem(53248)=0
1630 :     goto20
1640 : 90  mem(53248)=ic+1
1650 :     goto20
1660 : *left (x-pos)
1665 : 100 mem(2040)=194
1667 :     mem(2040)=193
1670 :     ic=mem(53248)
1680 :     ie=mem(53264)
1690 :     if (ie-0) 101,101,102
1700 : 101 if (ic-0) 20,20,105
1710 : 102 if (ic-0) 103,103,105
1720 : 103 mem(53264)=0
1730 :     mem(53248)=255
1740 :     goto20
1750 : 105 mem(53248)=ic-1
1760 :     goto20
1770 :     stop
1780 : *end of program
1790 :     end
```

Per il prossimo articolo ho deciso di prendere spunto da questo programma per realizzare il porting in Fortran del gioco **C-64 Road Racer**, pubblicato su **RetroLiPS**: https://www.retromagazine.net/c-64-road-racer-commodore-64-basic-v2/

Invito a … “Personaggio” per C64

di Eugenio Rapella

“Personaggio” è un vecchio programma per Vic-20 che era stato pubblicato su “Super Commodore” nel lontano 1985.

L’ho rispolverato recentemente e adattato al C64; magari può interessare a qualche lettore di RMW che voglia farsi … qualche risata (almeno questo sarebbe l’obiettivo).

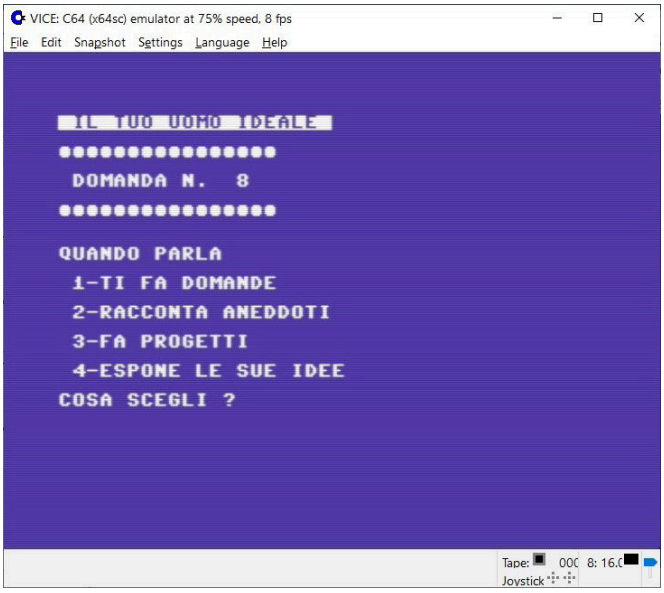
Il Commodore propone 20 domande a risposta multipla (vedi gli screenshot); alla fine una donna avrà trovato il suo uomo ideale, un uomo troverà il tipo di persona con cui vorrebbe identificarsi (può scegliere di rispondere “come sono” oppure “come vorrei essere”). Si può giocare anche “al contrario”: rispondendo con l’alternativa meno ambita si otterrà l’individuo meno consono alla propria personalità.

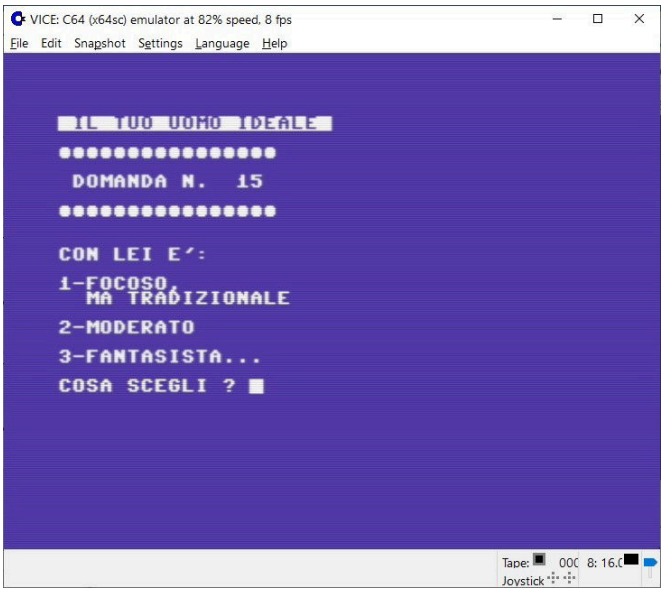
Per scaricare il programma, andate sul sito “ready64.org”; nella casella di ricerca, in alto a destra, scrivete “personaggio” e procedete al download.

Evitate di esaminare il listato: il risultato del test deve essere una … sorpresa!

Il C64 confronta la vostra serie di risposte con quelle che sono state associate ai vari personaggi (in modo del tutto personale) e propone il personaggio con il maggior numero di coincidenze.

Poiché il programma è del 1985, alcuni personaggi risultano “datati” e, naturalmente, non ci sono personaggi famosi degli ultimi … quarant’anni!





Niente vieta di aggiornare il programma:

1. Scegliete un gruppo di “personaggi” (uomini famosi, conoscenti amici, parenti), almeno una ventina (più sono, meglio è) e incolonnate i loro nomi su un foglio.
2. Date il RUN e, per ogni persona dell’elenco, scrivete a fianco del nome la risposta che associate alla persona.

Continuate così fino alla domanda n. 20. Al termine avrete qualcosa del tipo:

NONNA OLGA 11121123121111121112

ZIO MARCO 321…

3. Aggiungete (o sostituite, o variate) le DATA secondo il seguente formato:

888DATANONNA OLGA 11121123121111121112

(l’ordine di redazione di queste istruzioni non ha alcuna importanza).

4. Cambiate il valore di P nella linea 50 in modo che corrisponda al numero totale dei personaggi.
5. Salvate il programma così modificato e … giocate.

I personaggi ora presenti sono solo dei maschietti (ai tempi non si parlava ancora di quote rosa); naturalmente potete procedere diversamente.

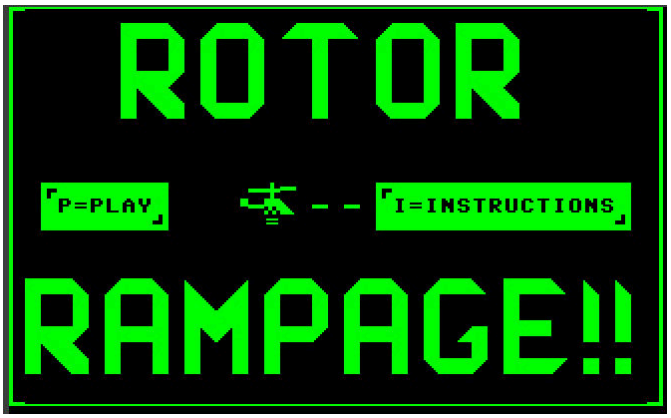
ROTOR RAMPAGE

Recensione del gioco ed intervista con gli sviluppatori
a cura di Gianluca Girelli

Abbiamo una missione, qui a Retromagazine World: preservare il passato e portarlo nel futuro. Il passato a cui siamo interessati è, ovviamente, quello che riguarda i sistemi retro in ogni forma possibile, dallo studio, riparazione e/o miglioramento dell'hardware al recupero, conservazione e/o sviluppo di nuovo software.

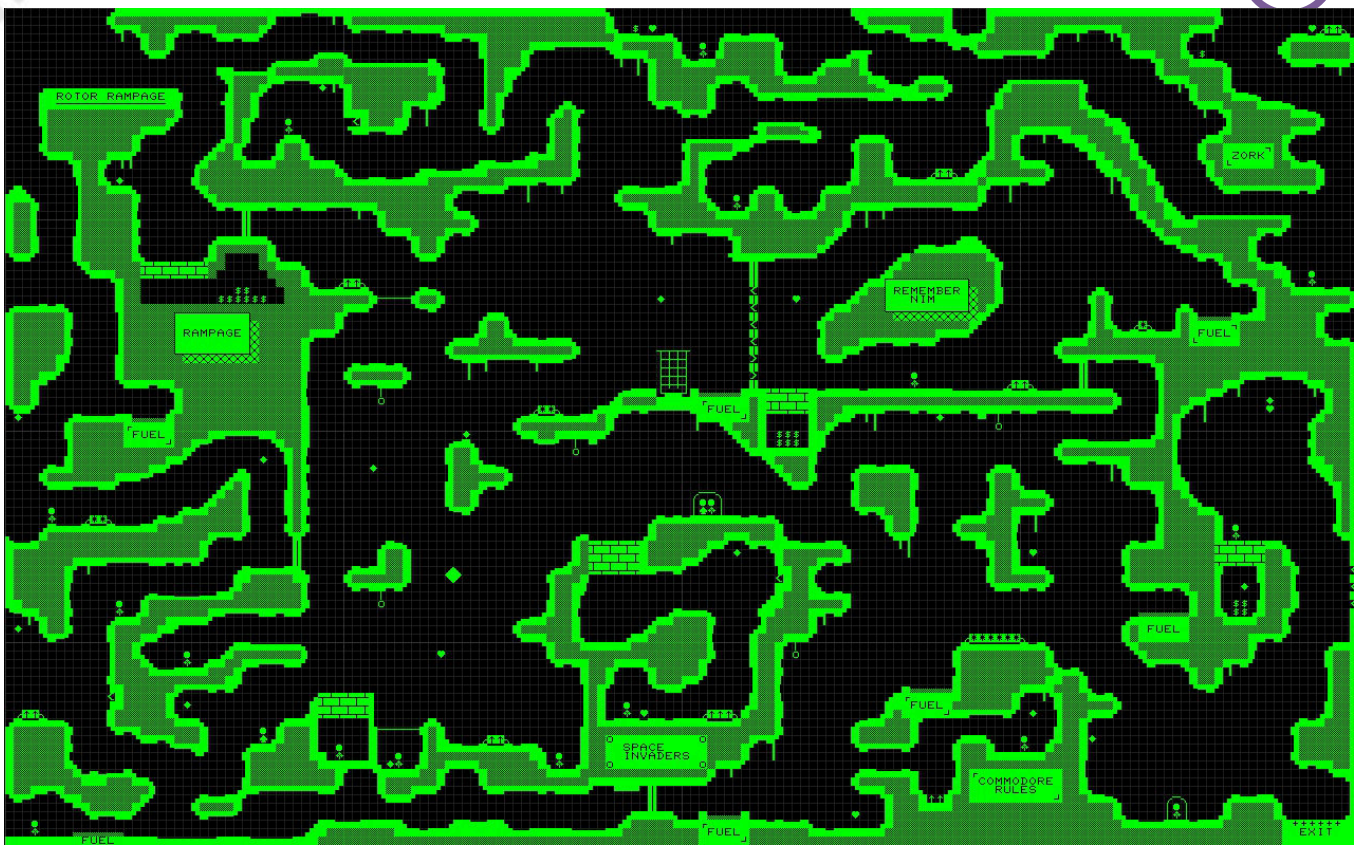
Di solito io e i miei colleghi redattori ci occupiamo (e scriviamo) di una manciata di sistemi, ma questa lacuna è dovuta solo al fatto che a volte ci manca una conoscenza approfondita di una macchina specifica. Ad esempio, la mia "arma preferita" è il Commodore C128, il mio primo computer e fedele compagno durante le scuole superiori. Da allora ho messo le mani su molti sistemi retro (Sinclair, Acorn, MSX e così via) ma, ovviamente, è il C128 quello che conosco meglio e perciò di cui scrivo. Eppure li amiamo tutti (anche quelli più “oscuri”) e, per preservare il passato, incoraggiamo e supportiamo chiunque possa aiutarci ad ampliare tali confini, soprattutto quando si tratta di parlare di macchine meno utilizzate o meno conosciute.

Siamo quindi lieti di presentare ai nostri lettori un nuovissimo gioco, Rotor Rampage, sviluppato appositamente per il meraviglioso Commodore PET da Dave Duke e Ken McIlveen.



"Rotor Rampage" è un videogioco fresco ed emozionante sviluppato appositamente per il Commodore PET, il classico computer che fece il suo debutto oltre 40 anni fa. Il gioco rappresenta una straordinaria miscela di nostalgia e innovazione, offrendo una nuova esperienza di gioco per gli appassionati di computer vintage. Personalmente, credo che questo sia un grande risultato e lo è ancora di più se si considera che il PET non è mai stato concepito per essere una macchina da gioco ma, piuttosto, una macchina per la produttività da ufficio.

Ho fatto una bella chiacchierata con Dave, qualche tempo fa, su cosa fanno questi ragazzi e su come il gioco è stato concepito e realizzato. In seguito anche Ken mi ha inviato



altri dettagli. Ecco cosa mi hanno detto.

RM: Ciao Dave, puoi raccontare ai nostri lettori qualcosa di te e di cosa fai?

Dave: Sono sempre stato appassionato di informatica e in particolare del PET, una meraviglia che la gente cominciava a buttare nella spazzatura quando uscivano nuove macchine ma che io continuo a collezionare e restaurare.

RM: Quando è iniziato tutto questo?

Dave: Ho iniziato a restaurare macchine circa vent'anni fa e da allora, penso di aver trovato e riparato circa 50 PET, che alla fine ho spedito in tutto il mondo, comprese Australia e Cina [Nota del redattore: Dave vive nel Regno Unito]. Gestisco anche un sito web, chiamato www.RetroMedics.com, dove si possono trovare informazioni utili sui PET, compresi i loro schemi.

RM: Raccontaci come hai iniziato a sviluppare il gioco.

Dave: Volevo creare un gioco moderno per il PET, tuttavia la sua memoria limitata significava che il programma avrebbe avuto funzionalità limitate come la maggior parte dei giochi dell'epoca. Ho deciso che doveva essere caricato da disco per essere il migliore possibile. A causa dei problemi di memoria e della velocità di esecuzione era chiaro che il codice doveva essere in linguaggio macchina e questo si è rivelato un problema significativo. Le routine di accesso al disco su un Commodore PET in linguaggio macchina erano impossibili da trovare, pertanto il progetto rimase congelato per diversi anni finché non incontrai Ken McIlveen, tramite un messaggio casuale in un forum online da un altro appassionato di 8 bit.

Si è scoperto che Ken aveva alcune routine dei primi anni '80 che potevano accedere ai dati del floppy disk dal linguaggio macchina su un PET.

Bingo, avevamo gli strumenti per costruire ciò che sognavo

da decenni.

RM: In che linguaggio è programmato il gioco?

Dave: Inizialmente ho utilizzato XCBASIC3 come strumento di prototipazione rapida, aggiungendo le routine in linguaggio macchina di Ken. Fu chiaro in breve tempo che per realizzare un gioco con tutte le caratteristiche che prevedevo sarebbe stato necessario scriverlo interamente in assembler. Ken è un mago dell'assembler 6502 ed era chiaro che tutto doveva essere portato su ML per ottenere la velocità e la funzionalità di cui avevamo bisogno.

Per un periodo di 5 mesi, Ken e io abbiamo lavorato sulle funzionalità e sul gameplay con l'obiettivo di inserire quante più funzionalità possibili in 32k di RAM. Penso che quando Ken ha finito di scrivere codice in assembler e di aggiungere le tante, tantissime funzionalità extra che continuavo ad aggiungere al progetto, gli sia rimasta solo una manciata di byte di memoria. C'è anche un messaggio nasconto in una delle mappe che fa appunto riferimento a ciò che in gergo viene definito "feature creep"!

Ken: Le parti in linguaggio macchina (e infine il resto del gioco) sono state codificate utilizzando il pacchetto

assembler CA65 all'unisono con VICE. Questi due software hanno probabilmente dimezzato i tempi di sviluppo. Le centinaia di ore di codifica e test che una volta avremmo dedicato allo sviluppo su un vero C64 sono state notevolmente ridotte, ma tieni presente che ci sono voluti ancora diversi mesi per completarlo. Durante lo sviluppo abbiamo anche tenuto presente che gran parte del codice avrebbe potuto essere riutilizzato per giochi futuri, quindi abbiamo mantenuto le cose in qualche modo generiche (internamente).

RM: È davvero impressionante e, come sviluppatore, intuisco come il gioco e l'intero processo dietro di esso si siano evoluti nel tempo. Potete dirci qualcosa di più su questa evoluzione?

Dave: "Rotor Rampage" è stato realizzato da due persone con la passione di mantenere vivi i giochi retro [Nota del redattore: lo adoro!] e nel gioco abbiamo a che fare con un elicottero che attraversa diversi livelli, mentre si fa strada sparando. Un motore fisico rudimentale esegue il rendering dei detriti che cadono, aggiungendo un ulteriore livello di divertimento. Ogni livello è composto da oltre 16.000 blocchi che possono essere caricati da disco o scheda SD. Il codice precarica banchi da 16KB, quindi ne carica un altro quando necessario. Il gioco presenta enormi livelli da attraversare, ognuno dei quali contiene molti riferimenti al passato. Inoltre, hanno tutti temi diversi, come: miniere, tombe egizie, mappe hardware interne del PET e così via. Infine, puoi creare e condividere le tue mappe utilizzando l'editor di mappe incorporato (di prossima uscita).

Ken: Il fattore trainante per me, come ha detto Dave, è stata la passione per il retro computing. Lavorare con questi sistemi per molti anni ha instillato in me un apprezzamento per la tecnologia del tempo. Ora, molti anni dopo, si tratta di tornare alle cose della vita che mi piace fare (ora che sono in pensione). Ecco perché abbiamo distribuito gratuitamente la versione PET. Originariamente questo era solo un progetto per passare i mesi invernali e non avremmo mai voluto che fosse altro. Il processo ci è piaciuto così tanto che le cose si sono evolute, e da lì e abbiamo deciso di realizzare una versione C64.

RM: Quali sono i requisiti per far funzionare il gioco?

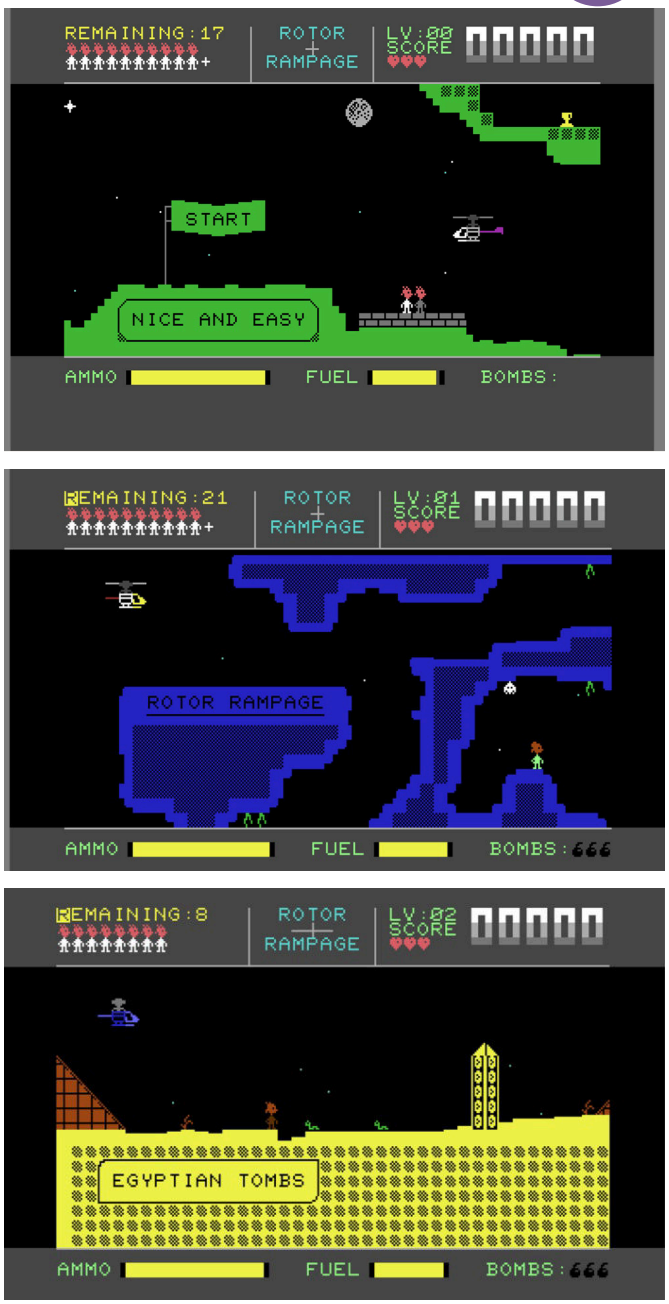
Dave: Il requisito principale è avere almeno 32k di memoria, sono supportati i Pet dal 2001-n al 4032, e finalmente ora a causa di molte richieste c'è una versione nativa a 80 colonne per il PET 8000+. Inoltre adesso supporta anche joystick e audio. La versione iniziale richiede il supporto del disco SD/Floppy, quindi assicuratevi che i vostri PET abbiano la ROM 2 o successiva.

RM: Che mi dite dei vostri piani per Rotor Rampage?

Dave: Partendo dalla versione PET originale, è ora disponibile anche una versione C64 completamente rielaborata. Con la memoria extra abbiamo aggiunto una grafica migliore, un suono migliore e una compressione dei livelli (il che significa che possiamo ottenere più livelli su un singolo floppy disk).

RM: Ora che avete completato le versioni PET e C64 di Rotor Rampage, su cosa lavorerete?

Dave: Stiamo ora lavorando su un gioco dungeon multiutente che supporta 3 client diversi, puoi giocarci utilizzando TELNET su Internet (questo significa anche qualsiasi PET con un client terminale), browser web, ma soprattutto un'app nativa C64 con immagini generate dall'intelligenza artificiale a schermo intero. Ogni dungeon può avere 1.000 utenti simultanei che giocano. Forse ci saranno anche personaggi IA interattivi nel gioco.



Immagini della versione Commodore 64

RM: Grazie mille per il vostro tempo e per questo gioco. Retro Magazine World continuerà a seguire i vostri progetti e vi aiuterà a spargere la voce. Mantenete viva la passione!

Link Utili

https://www.retromedics.com/

https://xc-basic.net/doku.php?id=v3:start

https://www.rotorrampage.com/

Intervista a Richard/TND, un maniaco del codice

di David La Monaca

Lo staff di RMW è lieto di avere l'opportunità di incontrare Richard del gruppo TND (The New Dimension), coder, programmatore, sviluppatore, musicista, grafico, redattore di diskmag, webmaster, blogger, ecc. Se pensate ad un ruolo nella "scena" del retrocomputing, e in particolare in quella relativa al Commodore 64, beh, Richard lo ha sicuramente ricoperto e lo ricopre tuttora. Un vero fan del computer a 8 bit più famoso del mondo e un grande sostenitore e divulgatore della Scena, allora cerchiamo di saperne di più sulle attività di Richard.

DLM: Ciao Richard e grazie mille per aver accettato il mio invito per un'intervista. Tutti i lettori e la redazione di RMW sono molto contenti di avere l'opportunità di farti alcune domande sulla tua esperienza nel mondo underground della computer scene targata C64.

RTND: Salve. È un vero piacere partecipare all'intervista. Sono felice di rispondere alle tue domande e di cercare di fornire quante più informazioni possibili sulla mia storia vissuta accanto al C64.

DLM: Puoi presentarti brevemente e raccontarci qualcosa della tua vita (ad esempio, di dove sei, la tua formazione scolastica, la tua crescita e le tue inclinazioni, i tuoi interessi personali al di là del lavoro, ecc.)?

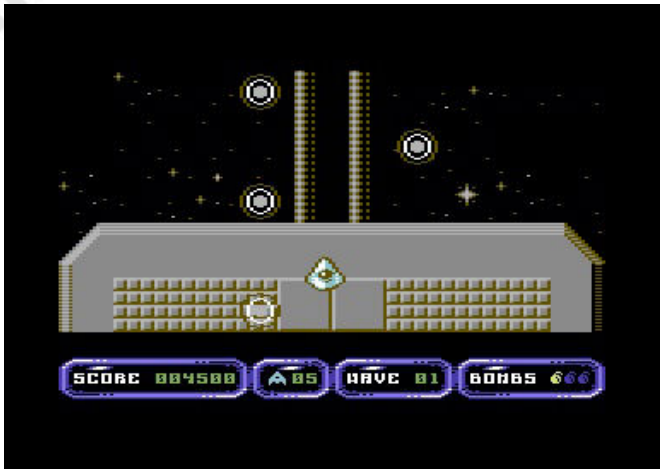
DMC Album - Un ampio selettore musicale contenente i fantastici brani di Richard.

RTND: Salve, mi chiamo Richard e vengo dal Regno Unito. In realtà non vorrei dare troppe informazioni sul mio conto, sulla mia istruzione, ecc. Come ogni buon inglese sono una persona molto riservata per natura! I miei interessi personali, al di là del mio lavoro, includono: videogiochi/retrocomputing, giocare a biliardo, freccette e fare lunghe passeggiate. Non ho molto interesse per la televisione di oggi a causa della scarsa qualità dei programmi che ho sempre subito intorno agli anni 2000.

DLM: Immagino che tu sia sempre stato un appassionato e un utente di computer fin da quando eri bambino. Cosa ti ha spinto sulla strada dell'informatica e qual è stata la tua prima esperienza con un computer? Il Commodore 64 è stato il suo primo micro computer?

RTND: Hai assolutamente ragione, e no, il Commodore 64 non è stato il mio primo micro! Il mio primo home computer è stato in realtà un Commodore VIC20 di seconda mano dotato di lettore di cassette C2N e anche una cartuccia di espansione RAM da 3K per riprodurre alcuni titoli "Bubble Bus". La maggior parte delle cassette che mi ero procurato erano per il VIC20 non espanso. All'epoca avevo solo 9 o 10 anni quando presi questo computer. Ricordo con piacere quanto divertimento provati giocando a diversi giochi per VIC20. Devo dire che i giochi che mi sono piaciuti di più in assoluto sono stati proprio quelli per il VIC20. Titoli come VIC Avenger, un gioco alla maniera di Space Invaders. Ricordo anche Triad, una variante di Galaxian. Mi piacevano anche Space Rescue, Duck Shoot e Super Breakout.

Nel 1990 era estremamente difficile acquistare i giochi



CruiserX79 - Veloce sparatutto a scorrimento verticale non fatto con SEUCK.

VIC20 nei negozi, nei mercati, ecc. E quindi non ne avevo molti, probabilmente 15-20 in tutto. Andai in alcuni negozi di materiale elettrico e mi piacque molto l'aspetto dei giochi per Commodore 64 che erano esposti. Così desiderai un Commodore 64 come regalo di compleanno. E nel dicembre del 1990 aprii il mio regalo di compleanno e si trattava proprio di un Commodore 64 accompagnato da un Datassette e da alcuni giochi. Nei due anni successivi ho ricevuto alcuni accessori (unità disco, cartuccia Action Replay MKVI, stampante a colori Star LC100 e mouse con varie utility software di supporto).

Durante quel periodo ebbi anche l'occasione di giocare con un Amstrad CPC, poiché di tanto in tanto andavo a trovare uno dei miei zii che aveva un computer Amstrad CPC. Comunque, tornando al VIC20, ricordo che adoravo veramente quella macchina ai tempi, ma poi, alla fine degli anni '80/inizio anni '90, era davvero difficile trovare e acquistare nuovi giochi per il VIC20.

Durante la mia infanzia ricordo di aver anche usato il BBC Micro Computer alle scuole elementari e medie, giocando a titoli educativi come POD (da non confondere con il gioco per C64 di Mastertronic) e Symetry. Ricordo di aver giocato a Chuckie Egg a scuola, in prossimità delle vacanze scolastiche stagionali. Era un bel divertimento giocarci, così, anni dopo, negli anni '90, comprai Chuckie Egg per il mio Commodore 64.

DLM: Come hai iniziato a esplorare il tuo C64, oltre a giocarci? Hai trovato fin da subito interesse nella programmazione e nella scoperta del funzionamento intimo della macchina?

RTND: Ho iniziato a lavorare sul C64 oltre a giocarci utilizzando lo Shoot'Em Up Construction Kit (SEUCK), ancora prima di programmare. SEUCK era in dotazione con il mio Commodore 64. Ho creato tantissimi giochi utilizzando quell'editor. Mi piaceva molto sviluppare giochi con il SEUCK. La mia grafica e il mio design erano piuttosto scarsi per un adolescente. Tuttavia, era molto divertente creare giochi con questo strumento.

La fase in cui mi sono avvicinato alla programmazione è iniziata nel 1993, quando sono andato a trovare un mio amico dopo la scuola. Mi mostrò alcuni programmi che aveva realizzato in BASIC. Erano cose molto semplici, ma nel 1994 riuscii anch'io a programmare il mio primo gioco in BASIC, che chiamai "Cidcom". Era un gioco in cui c'erano dei numeri che lampeggiavano sullo schermo e dovevi premere la barra spaziatrice per impedire che i numeri cambiassero. Non lo pubblicai perché fino al 1995 non ero a conoscenza della scena del software di Pubblico Dominio.

Il 1999 è stato l'anno in cui ho iniziato a imparare il codice in linguaggio assembly attraverso due fanzine cartacee per C64, Commodore Scene e Commodore Zone. Entrambe le riviste erano alternative valide a Commodore Format ed erano anche fornite di allegati software. Su alcuni di questi era pubblicato un Turbo Assembler su nastro, insieme ad alcune routine di codice sorgente dotate di relative spiegazioni. Ricordo anche di aver imparato qualcosa da riviste come Coder's World e forse da altre, tra la fine degli anni '90 e l'inizio del 2000. Ho dovuto fare affidamento su risorse limitate finché non ho avuto accesso regolare a Internet nei primissimi anni 2000 grazie ai servizi a banda larga :).

DLM: Qual era la configurazione del tuo primo computer (comprese le periferiche, le memorie di massa, le espansioni e gli accessori)?

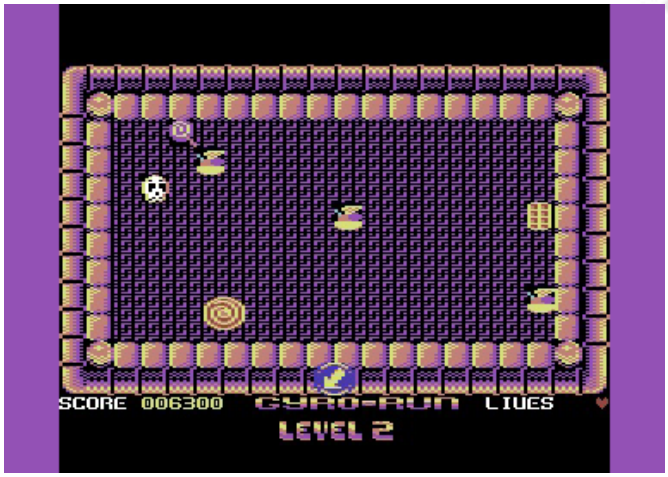
RTND: La configurazione del mio primo computer (il VIC20) era molto semplice. Avevo un televisore in bianco e nero (poi cambiato con un piccolo televisore a colori Saisho CRT), un Datassette, il classico joystick ATARI e un'unità di espansione RAM da 3K insieme ad alcuni giochi. Tra questi ricordo: Awari, Jupiter Lander, Mars Mission, Alien Plague, Duck Shoot, The Wizard and the Princess e un paio di cassette di VIC Tape Computing, che contenevano alcuni giochi come “Pennies from Heaven”.

DLM: Cosa ti ha spinto ad avvicinarti alla grafica, alla musica e agli effetti sonori del computer e a decidere di provare a progettare opere solo tue?

RTND: Giocai a tante produzioni impressionanti di cui ero molto soddisfatto, soprattutto i giochi per lettori allegati alle riviste, pubblicati su alcune cassette di Commodore Format, Commodore Force e Zzap!64. Questi giochi mi hanno in qualche modo ispirato a provare a sviluppare qualcosa di mio per il C64.

Quanto alla musica, tra il 1994 e il 1995 e i primi anni 2000 ho raccolto molti dischi e nastri di utilità di pubblico dominio. Alla fine disponevo di diversi editor musicali realizzati da vari membri dei gruppi della Demo Scene. Così ho cominciato a sperimentare con questi editor/tracker prima di creare i miei strumenti software. Il primo editor musicale che ricordo era in dotazione con il mio C64 ed era piuttosto elementare. Era “The Music System” di Firebird. Si trattava comunque di un editor musicale divertente. Gli editor/tracker PD come DMC, Music Assembler, Music mixer, ecc. avevano dei suoni in regalo incorporati nell'editor e con questi si potevano creare colonne sonore davvero impressionanti per l’epoca.

DLM: Come hai imparato a programmare e qual è stato il tuo primo programma, se lo ricordi?



Game Gyro Run - Un gioco di trottole molto strano, che vi farà girare la testa.

RTND: Il mio primo programma per C64 è difficile da ricordare e purtroppo non ce l'ho più. Ho imparato a programmare nel 1998/1999, quando nel 1995 mi mostrarono una fanzine chiamata "Commodore Zone" (dopo che Commodore Format aveva chiuso definitivamente le sue pagine nel settembre del 1995). La fanzine in formato A5 veniva fornita con un allegato (opzionale in formato disco o cassetta). Nei numeri successivi c'era un articolo chiamato "The Hex Files". Si iniziava con le nozioni di base sulla visualizzazione di oggetti sullo schermo, la creazione di testi a scorrimento, la realizzazione di semplici demo di immagini, fino alla creazione di giochi completi. Con l’aiuto di Internet, contattai anche altre persone della scena che mi aiutarono a volte quando cercavo di imparare a creare nuove cose.

DLM: Com’era composto l'intero sistema che utilizzavi per sviluppare e testare i primi progetti delle tue produzioni (giochi, demo, tracce SID, immagini grafiche)?

RTND: Prima di scoprire gli strumenti di sviluppo tipo cross-assembler, usavo un C64C con lettore disco 1541-II e alcuni dischi di pubblico dominio che compravo nelle librerie PD del Regno Unito per meno di 2 sterline. La maggior parte di essi conteneva tools e utility per la programmazione. Usavo principalmente Turbo Assembler V5 per il codice e una cartuccia Action Replay per usare

il monitor del codice macchina. Utilizzavo anche editor musicali come Music Assembler V1.0 del Dutch USA Team, DMC di Graffity e alcuni altri software per comporre. Gli sprite venivano disegnati con Sprite Editor V1.3 di Faces, mentre i caratteri venivano disegnati con Dunex Char Editor V2.0. Quando progettavo le schermate di gioco, in pratica caricavo il charset e modificavo la schermata manualmente sullo schermo BASIC. Era un lavoro piuttosto faticoso. Dopo aver terminato il gioco, usavo un packer e un cruncher (queste operazioni richiedevano parecchio tempo).

DLM: Quando hai iniziato a far parte della scena C64? Come ti è venuta l'idea? Come hai iniziato a collaborare con altri amici e scener? Come comunicavate tra di voi a quel tempo?

RTND: Sono entrato per la prima volta nella scena C64 nel 1995, quando ho inviato del materiale per Binary Zone PD (Coloured, Nyaaaah! e Sector 5). Non mi rendevo conto che stavo facendo un passo avanti verso qualcosa di nuovo. Questo fino a quando non è arrivato il boom di Internet. Il primo gruppo della Scena a cui mi ero unito fu Civitas su invito specifico di Zeitgeist (che purtroppo è scomparso misteriosamente dall'ultimo contatto avuto nel 2006). Ho dato una mano al codice di demo, intro e alla creazione di musica, persino qualche gioco. Quando Internet è diventata ancora più grande sono stato contattato da molte persone della Scena e ho collaborato con diversi progetti passati e ancora vi collaboro oggi.

Mix Box Music Disk - Un buon mix di brani diversi per le vostre raffinate orecchie.

DLM: Hai seguito dei corsi di informatica per iniziare il tuo percorso nel campo dell'elettronica e dell'informatica, e se sì quali erano e quanto tempo hai investito? Oppure, come molti designer/programmatori dei primi anni Ottanta, eri un autodidatta del coding?

RTND: Tra il 1995 e il 2000 ho seguito alcuni corsi di informatica. All'inizio si trattava di un primo diploma nazionale e di un diploma nazionale superiore in informatica. Alcune parti del corso comprendevano la programmazione. Ho usato GWBasic/Q-Basic, Visual Basic e C++. Sebbene il C++ fosse il linguaggio di programmazione più "antipatico", ho comunque completato i corsi e li ho superati a pieni voti. Quindi nella vita reale ho investito 5 anni all'università con i corsi di informatica. La mia materia preferita era proprio la programmazione. Ricordo che una volta, prima del Natale 1995, ci fu fatto una specie di regalo, ovvero un paio di listati GWBASIC completi che consistevano in giochi da digitare. I giochi erano Concentration e Artillery Duel.

Per quanto riguarda la programmazione sul C64, sono stato una specie di autodidatta. Tuttavia, a volte ho dovuto fare affidamento all'aiuto di membri della Scena per alcuni problemi specifici che ho incontrato - o che non avevo mai visto prima. Per fortuna c'erano sempre alcune persone molto disponibili che mi aiutavano con le insidie che incontravo e che cercavo di superare imparando molto allo stesso tempo.

DLM: Come si svolgeva il tuo processo di sviluppo di un nuovo progetto o produzione all'epoca? Per esempio, abbozzavi disegni e schemi per seguire un concetto iniziale, parlavi con un team della trama, traevi ispirazioni per la colonna sonora, le immagini, ecc. da altri titoli?

RTND: Niente di tutto questo. Non volevo sprecare carta e inchiostro per progettare e sviluppare un nuovo gioco. Piuttosto avrei creato una struttura utilizzando la grafica

grezza direttamente sul computer. Poi, in un secondo momento, avrei migliorato un po' le cose.
Quando si tratta di fare musica, sperimento con gli editor musicali e i tracker finché non ottengo i brani giusti che ritengo adatti alle mie produzioni.

DLM: La tecnologia e molti strumenti di sviluppo sono sicuramente cambiati nel corso di questi anni, ma per quanto riguarda la parte creativa del tuo lavoro, affronti ancora il processo nello stesso modo in cui hai lavorato per le tue produzioni più recenti?

RTND: Concordo con te che gli strumenti di sviluppo sono cambiati nel corso degli anni. Anche se la maggior parte del tempo, uso tool di cross-development, come C64Studio, CBMPRGStudio per la programmazione, Cuneiform, Char Pad, Multipaint e Sprite Pad per la grafica, ed Exomizer per la compressione. Uso anche l'emulatore VICE per i test. In questo modo il processo di sviluppo, compressione e test del gioco è più breve rispetto a quello che si faceva un tempo.

Usavo una cartuccia freezer come Action Replay, dischi di utilità di pubblico dominio per creare e sviluppare grafica e musica e Turbo Assembler per programmare il software. Mi sono divertito molto con questi strumenti. In realtà, di tanto in tanto li uso ancora :-).

DLM: Qual è stato il più grande ostacolo tecnico che hai superato durante la progettazione/programmazione/test delle tue produzioni per le macchine a 8 bit (C64)?

Sicuramente la temporizzazione dei raster e la correzione dei bug sono argomenti che mi hanno dato filo da torcere. Questa fase dello sviluppo di codice l'ho davvero odiata. A volte si tratta di qualcosa di molto semplice, altre volte di una vera e propria sfida. Ad esempio, quando ci si sposta per cercare di eliminare lo sfarfallìo, si passa alla linea raster successiva. Più codice si aggiunge e più cambia ancora la temporizzazione. Trovo che la cosa migliore da fare sia lasciare la temporizzazione raster per una demo/intro o qualcosa del genere come ultima parte prima dell'effettiva pubblicazione.

Una vera sfida per me, che non ho mai affrontato davvero, è lo scorrimento a doppio buffer nei giochi. Soprattutto perché la maggior parte delle volte non ho abbastanza memoria per farlo. La RAM di solito viene assorbita da altri dati. Questo è il motivo per cui la maggior parte dei miei giochi usa solo tre colori se utilizza lo scorrimento verticale o orizzontale della mappa di gioco. Amazon Tales, per esempio, usa più colori, ma non utilizza il double-buffering.

DLM: Qual è stato/è il tuo gioco preferito per C64? Trovi ancora un po' di tempo per giocare ai videogiochi?

RTND: Per quanto riguarda i giochi preferiti, il mio gioco preferito di sempre per il C64 è "Mayhem in Monsterland". Ricordo di aver provato la demo giocabile del gioco nel 1993, quando Commodore Format lo regalò sul nastro Power Pack a doppia cassetta del numero 37. Alla fine ho acquistato la versione completa su disco. Rimasi molto colpito dalla presentazione generale, dalla grafica e dal concetto.

C'è anche un altro mio gioco preferito per C64 che è una vera sfida da giocare. Si tratta naturalmente di Armalyte. Per me è forse uno degli shoot'em up su C64 più difficili

Rocket & Roll - Vivi il viaggio più divertente della tua vita in questa avventura arcade fatta con SEUCK.

Starfysh - Da quando le corsie dei taxi attraversano la galassia?

di tutti i tempi. Non riesco nemmeno a superare il primo livello! Nonostante questo, l'ho rigiocato tante e tante volte mettermi alla prova.

Riesco a ritagliarmi un po' di tempo per giocare ai videogiochi di solito nei giorni feriali, quando la TV è dominata da notizie deprimenti e programmi di attualità, soap opera e film drammatici. Altre volte, navigo su Internet o faccio cose per la C64/Scene come programmare, comporre musica, lavorare per la disk magazine Scene World, ecc.

DLM: Qual è stata/è la tua colonna sonora/chiptune preferita per C64? Qual è la tua "top 3" delle tracce originali SID mai pubblicate? Trovi ancora un po' di tempo per seguire le ultime produzioni?

RTND: La mia colonna sonora preferita è Cybernoid 2 di Jeroen Tel. In realtà non ho una top 3 dei brani SID preferiti. Di tanto in tanto trovo il tempo di seguire le ultime produzioni grazie a Internet.

DLM: Parlando di brani SID, qual è il tuo artista/musicista/compositore preferito di sempre per il C64?

RTND: Sicuramente Jeroen Tel perché le sue melodie nei giochi per C64 sono orecchiabili e memorabili.

DLM: Ora una domanda che tutti vorrebbero farti come esperto coder per C64: quali strumenti usi attualmente per creare giochi, demo, applicazioni, ecc. e qual è la configurazione (usi un Mac o un PC)?

RTND: Gli strumenti che uso regolarmente per creare grafica sono SpritePad V2.0, CharPad V2.7.6 (entrambi in edizione gratuita), GangEd, PixCen, Multipaint, Cuneiform, Ultrafont PC. Per la programmazione/sviluppo CBMPRGStudio, C64Studio, a volte il semplice Notepad++, KickAssembler, ACME cross-assembler, TMPx. Per la compressione uso regolarmente Exomizer V3.1.2 o TSCrunch.

DLM: Quale editor e quali strumenti utilizzi quando componi brani per vecchie macchine come C64, Amiga o console? Ti piace usare ancora il vecchio hardware?

RTND: Ci sono alcuni compositori che uso per fare musica su C64. L'editor musicale più comune che uso è sul C64, ed è DMC V4.0 oppure DMC V5.0. Recentemente ho usato EMS V7.03. In effetti lo sto usando su un gioco per C64 che ho sviluppato durante il periodo invernale e che spero di rilasciare nell'aprile del 2024 (se il gioco verrà completato). Sul PC utilizzo regolarmente Goat Tracker V2, ma occasionalmente potrei utilizzare anche CheeseCutter e SidFactory 2.

DLM: Immagino che tu possieda una collezione di C64 stock (cioè tutte le versioni: C64 "breadbin", C64c, C64g, C128) per scopi di test o solo per nostalgia. Mi sbaglio? Sei anche un collezionista di retrocomputer, non solo di marca Commodore?

RTND: In realtà non sono un grande collezionista di computer retro. Tuttavia, ho collezionato una discreta quantità di giochi per C64 su cassetta. Si tratta principalmente di giochi economici e nastri di allegati alle copertine delle riviste (Commodore Format Power Pack, Zzap! 64 Mega Tape, Commodore Force Reel Action). Sono sempre stato legato al C64 e ne ho avuti 5 in passato,

di cui tre sono ora difettosi e uno è stato distrutto da un bambino in visita a casa mia. Erano tutti C64C.

La motherboard del mio C64C più longevo è stato sostituito da un Ultimate64 nel vecchio case C64C a causa di problemi di alimentazione. Ricordo che questo C64C mostrava una debole e grande linea nera che scorreva sullo schermo e la mappa caratteri si confondeva. Quindi, come ultima risorsa, l'Ultimate 64 è stato un ottimo sostituto per me.

Tra le altre tecnologie retro che possiedo ci sono un computer C64 Mini e un C64 Maxi (sul quale posso anche giocare a giochi VIC20 memorabili). Ho anche una mini console NES Classic, un Evercade VS prima edizione per i giochi retro non legati al C64. Ho anche una macchina arcade personalizzata che mio fratello ha realizzato per il mio 40° compleanno qualche anno fa.

DLM: All'inizio lavoravi da solo sul C64, in squadra o semplicemente ti consultavi con altri coder per ottenere un particolare risultato o per risolvere un bug nel tuo codice? Cosa preferisci in generale, lavorare in un team o da solo?

Solitamente provavo a sviluppare le mie produzioni da solo, ma da quando la comunità Internet del C64 è diventata grande, penso che sia meglio lavorare e collaborare come una squadra, perché con una buona squadra ci può essere molta più qualità in una produzione. Mi piace sviluppare per il C64 per divertimento, non per soldi. È sempre una buona idea collaborare con altre persone. Loro tendono a testare il mio materiale e a segnalare qualsiasi bug da risolvere, oppure mi consigliano i miglioramenti che possono essere realizzati in una produzione C64 in base all'esperienza comune.

DLM: Sono sicuro che negli ultimi anni hai lavorato molto duramente ai tuoi progetti, ma anche che ti sei divertito molto nel farlo. Qual è il momento o la storia più divertente/strana che hai vissuto mentre lavoravi alle tue produzioni?

RTND: Sviluppare produzioni per C64 è stato davvero molto divertente. A volte può diventare un grattacapo quando affronta la fase di correzione dei bug all'ultimo minuto, ma il risultato complessivo di solito è piuttosto buono (o cattivo).

A proposito del momento più divertente dello sviluppo di un nuovo gioco per C64: il momento più divertente (e forse esilarante) è accaduto nel 2022. Ho visto un sondaggio su Twitter pubblicato dal Commodore Format Archive. C'era scritto:

Votate questo importante sondaggio:
- Dick Tracey
- Cisco Heat
- Tasered in the Crotch

Star Hawx - Questi malvagi uccelli spaziali vogliono conquistare la Galassia. Andate a dar loro da mangiare una torta laser.

Snake vs Bomb 2 - Questo serpente è affamato di frutta, ma potrebbe andare incontro ad una fine esplosiva?

Dopo la chiusura del sondaggio, il risultato è stato rivelato. Per farmi una risata, ho deciso di creare un piccolo gioco chiamato "Tasered in the Crotch", una sorta di tiro a segno. Nel gioco si controlla una squadra di agenti di polizia che devono colpire con il taser i rapinatori (sono vestiti con maglioni a righe e portano sulle spalle sacchi col bottino) ed evitare di colpire i cittadini comuni. Dopo il lancio del gioco, il feedback sulla produzione è stato sorprendente. È buffo come una cosa fatta per ridere si sia trasformata in un capolavoro divertente. Quando ho parlato di questo gioco alla mia famiglia mentre ero fuori casa, ne abbiamo riso tutti.

DLM: Ripensando al punto di partenza, c'è qualcosa che rimpiange di una singola produzione (musica o programma/gioco/strumento/applicazione) o di qualsiasi altro dettaglio? Faresti qualcosa in modo diverso se potessi?

RTND: Sub Hunter era un grande gioco, e ci sono voluti più di 4 anni per svilupparlo. Sfortunatamente per alcuni nella scena del cracking del C64 era mal programmato. Il gioco non era così tecnico, ma ricordo di aver visto il titolo craccato e modificato con trainer da un falso gruppo di cracking, con un messaggio sulla produzione indirizzato a me che diceva di imparare a sviluppare codice per bene. Il codice era stato da loro classificato come “spaghetti code” (si legge nell’intro che hanno messo davanti al gioco). Chiaramente non ne ero molto contento.

Tuttavia, se potessi tornare indietro nel tempo rispetto a quel progetto di gioco. Avrei forse modificato il codice in modo da utilizzare dati auto-modificanti, invece di ripetere costantemente lo stesso codice e gli stessi interrupt per ogni livello. Questo è stato probabilmente uno dei miei più grandi progetti di gioco a cui ho lavorato. Ad un certo punto mi sono perso, ma alla fine sono riuscito a portare a termine il progetto.

DLM: Quando è nata l'idea di TND, come gruppo di produzione e come sito web che hai costruito per pubblicare tutte le uscite e anche alcuni tutorial di programmazione e tecnologia?

RTND: Il sito web di TND (se ricordo bene) potrebbe essere nato nel 2000 o nel 2001. Inizialmente era stato creato per mostrare le mie produzioni. In seguito si è trasformato in una sorta di comunità che invita le persone a condividere i propri contenuti e contribuire al sito web. È stato molto bello essere contattato con giochi C64 da condividere sul mio sito. Quando arrivarono i giochi SEUCK, volevo che la gente non aspettasse a lungo il caricamento del disco. Così caricavo il gioco, aggiungevo della musica (avvisando prima l'autore) e poi comprimevo i giochi in eseguibili da caricare in C64 BASIC. Gli autori ne erano piuttosto soddisfatti.

In seguito, ho ampliato il sito web di TND per farlo diventare qualcosa di leggermente diverso. Le esercitazioni di programmazione e di tecnologia erano divertenti da preparare, compresa la Scuola SEUCK. Non ricordo quando ho iniziato a gestire le esercitazioni di programmazione. So solo che la rubrica “Assemble It” è iniziata come un piccolo tutorial di sviluppo per creare intro, piccole demo e piccoli giochi mentre stavo ancora imparando a programmare in assembly sul C64. Gli esempi erano originariamente basati su Turbo Assembler.

Nel 2018 è stata lanciata la mini replica del C64 (il

"theC64"). Ho provato la mini console nell'aprile 2018, prima di preordinare una Ultimate 64 nel giugno 2018 (quando il mio Commodore 64 ha cominciato ad avere problemi). Con l'insorgere della comunità retro che si preparava a creare nuovi giochi per questa console, ho deciso di espandere "Assemble It" e di renderla ancora migliore di prima. Questa volta "Assemble It" era basata su IDE moderni piuttosto che su Turbo Assembler. Poiché all'epoca usavo molto ACME Assembler, decisi di usare C64 Studio come strumento principale per scrivere il codice e anche strumenti di cross-development come Char Pad e Sprite Pad per la grafica, Goat Tracker per la musica ed Exomizer per la compressione.

La nuova sezione "Assemble It" non riguarda solo la creazione e lo sviluppo di un nuovo gioco per C64. C'è un po' di programmazione introduttiva, creazione di demo, masterizzazione di dischi, nastri e cartucce e infine una sezione con i miei giochi precedentemente rilasciati completi di codice sorgente.

La sezione SEUCK School si basa sul miglioramento dei giochi creati con Shoot'Em Up Construction Kit. Si va dall'aggiunta di musica ai front-end e al gioco, fino alla realizzazione di giochi SEUCK completi, come la creazione di un nuovo front-end, l'aggiunta di potenziamenti e molto altro. Alcuni fanatici di SEUCK hanno persino contribuito a fornire alcuni suggerimenti per il miglioramento di SEUCK. Ho inserito anche un servizio speciale sul motore di SEUCK Redux.

DLM: Tra le tue numerose pubblicazioni ci sono molti giochi SEUCK. Quando hai incontrato per la prima volta questo strumento di sviluppo e perché ti è piaciuto così tanto da esplorare tutte le possibilità disponibili per creare giochi?

RTND: Ho incontrato per la prima volta il SEUCK nel 1990, perché era incluso nel pacchetto "Light Fantastic" del mio Commodore 64C. Mi piaceva molto perché all'epoca non dovevo fare alcuna programmazione. Creare un gioco

Scene World 32 DiskMag - Una rivista su disco PAL/NTSC, spontaneamente creata da appassionati di C64 / retro computing di lunga data.

Sub Hunter - Una vagonata d'azione in questo superbo sparatutto disseminato di nemici sotto il mare

utilizzando il kit era piuttosto semplice. SEUCK ha fatto parte della mia giovinezza. Mi ha anche aiutato a diventare un progettista di giochi per C64, anche se la maggior parte del tempo la passo a programmare.

DLM: Hai lavorato almeno su due diversi diskmag, "GameOverview" e "Scene World", quest'ultimo probabilmente il più famoso diskmag per C64 insieme a Vandalism News.
Eri un lettore di diskmag prima di entrare nello staff di un diskmag e diventare redattore?

RTND: Non mi sono occupato attivamente di GameOverview, ma ho contribuito alle uscite con della musica. Scene World è stato l'unico diskmag per il quale ho scritto principalmente materiale. Sono stato anche un lettore di alcuni diskmag della scena prima di entrare come redattore di Scene World. Mi piacevano soprattutto i diskmag che contenevano software bonus da caricare ed eseguire o da cui imparare il codice. Scene World era un piccolo diskmag prima di diventare qualcosa di grande su Internet.

Con quest'ultima domanda termina l'intervista a Richard/TND. Ancora una volta, lo ringraziamo per aver dedicato il suo tempo a rispondere alle nostre domande. Speriamo che i lettori si siano divertiti a leggerla e possano trarre ispirazione dalle numerose attività di Richard e dalla passione che mette in tutti i suoi lavori.

Riferimenti

TND
- https://tnd64.unikat.sk/

SWO
- https://sceneworld.org/

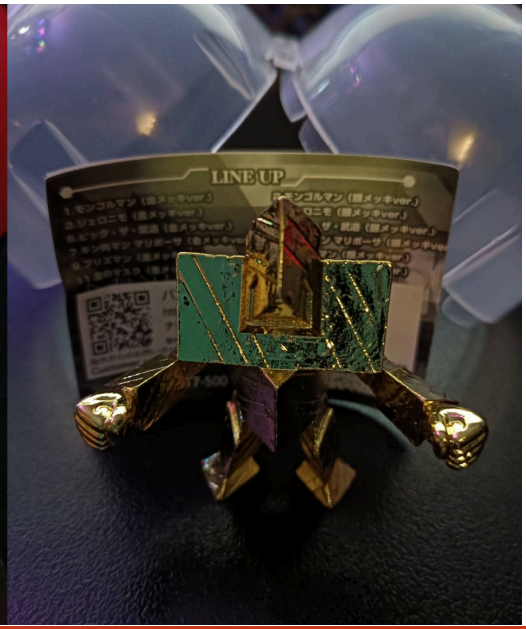
RCB (pagina CSDB di Richard)
- https://csdb.dk/scener/?id=1201

Richard's GitHub page
- https://github.com/RichardTND

Giappone 24^ puntata: Gacha...pon!

di Michele Ugolini

Quante volte da bambini abbiamo esasperato i nostri genitori durante le famose giornate di shopping dedicato agli articoli del nostro vestiario?
Quante volte da bambini siamo stati esasperati dai nostri genitori durante tantissimi acquisti, apparentemente inutili, dentro i centri commerciali?
Quante volte da bambini ci siamo lamentati per andare dal barbiere X, nel negozio Y, nella farmacia Z, etc.. etc.. etc..
Possibile che non sia mai esistita una soluzione a questo drammatico, mondiale ed ubiquitario problema dei bambini, innocenti anime, oppresse dai settimanali obblighi famigliari?
Possibile che nella mente dei nostri santissimi e pazientissimi genitori non sia mai stato espresso il desiderio di una soluzione a questo flagello mondiale? Perchè? Perchè non esiste una soluzione a tutto ciò? Pianti e capricci di bambini trascinati a forza nei centri commerciali... dal barbiere... dal dentista...
Siamo realmente sicuri che, a questo problema mondiale, non esista alcun tipo di soluzione, in nessuna parte del globo?
Cari lettori, sono da poco rientrato dal Giappone e, come potrete immaginare, sto proprio per parlare della soluzione a questo drammatico flagello mondiale: salvare i bambini di tutto il mondo dagli odiosi obblighi degli acquisti famigliari.
Un tutorial per salvare i timpani dei genitori dagli strilli dei loro adorati figli! Nonchè un suggerimento per tanti negozianti!
Certamente, ho esagerato il concetto, ma se ci pensiamo bene, la temperatura è il totale della quantità di calore moltiplicata per il volume di una massa. Quindi per analogia, se moltiplichiamo tantissimi piccoli pianti (poco calore) per un numero enorme di bambini tristi (massa), sicuramente raggiungiamo un termometro di insoddisfazione mondiale degno di nota: un termometro che misura la febbre alta!
Scaviamo dentro i nostri ricordi, quando eravamo bambini, saltuariamente i nostri genitori ci regalavano qualche piccolo, economico, giocattolo per portarci a messa o dal dentista o nei negozi... senza troppi strilli. I più fortunati potevano passare in edicola a comperare una amatissima cassettina del Commodore 64. Soprattutto noi italiani abbiamo veramente amato queste cassettine. Misteriose cassettine con una descrizione dei giochi degna di "gloriosi propositi" spesso reincarnata dentro pochi, scarni, pixel! Se scaviamo ancor più profondamente dentro i nostri ricordi, probabilmente in moltissime città e paesi in periferia possiamo ricordare che fuori e dentro le edicole, nonché presso altre posizioni più o meno improbabili, esistevano delle macchinette che distribuivano chewingum e pallette di plastica contenenti qualche oggetto misterioso! Ci sono anche ora. Spesso ignorate e ingiallite negli angoli dei giornalai ed alcuni tabaccai!
Il fenomeno di queste pallette non è molto sentito in Italia, mentre in Giappone è una autentica moda, ritornata in voga in questi ultimi dieci anni, dopo un breve sonnellino attorno al 2000.
In Giappone questi distributori di pallette, contenenti qualsiasi oggetto la nostra immaginazione possa "credere di immaginare", sono in ogni punto strategico del turismo, dei centri commerciali, a fianco barbieri, negozi di vestiti dei bambini, lungo le vie con file lunghissime di macchinette, negli aeroporti, fuori da negozi che vendono merce particolare da collezione, etc...
Questa macchinette, quasi mai dotate di energia



elettrica, sono basate unicamente sul principio meccanico di ricevere monete da 100 yen per poi emettere una pallina tramite la rotazione manuale di una manovella rotonda. Il nome Gachapon è proprio dato dal rumore di questa manovella che viene emesso mentre la giriamo con la nostra mano, dopo aver inserito il corretto numero di monete. Un suono plasticoso simile appunto a "gachagachagacha".
Dopo aver girato la manovella viene sbloccato il serbatoio di sfere contenute e cade una palletta fuori dalla macchinetta, nell'apposito recipiente, con il classico tonfo "pon". Ecco svelato il termine Gachapon. I giapponesi spesso usano questa logica onomatopeica per arricchire molti sostantivi e verbi, per esempio "dokidoki" dimostra la tensione emotiva del batticuore appunto tramite la ripetizione del suono del battito stesso.
I principali marchi registrati di questi Gachapon sono Takara Tomy e soprattutto Bandai. Sono entrambi marchi talmente famosi che ogni presentazione è superflua, oltretutto Takara Tomy ha recentemente spedito sulla luna dei mini rover (SORAQ) sferici che sono riusciti a svolgere svariate missioni con successo.
Peccato che il lander principale sia atterrato in posizione obliqua. Ad ogni modo i giapponesi imparano presto dai propri errori e sicuramente la prossima missione sarà più proficua.
Ritorniamo ai nostri centri commerciali giapponesi con aree dedicate a vasti corridoi pieni zeppi di queste macchinette. Possiamo immaginare queste vie affollate da bambini rumorosi! No, nulla di più sbagliato! Per ogni bambino silenzioso che sceglie con cura dove spendere le proprie monete, ci sono almeno due madri che chiacchierano scegliendo la propria macchinetta, magari commentando qualche aneddoto buffo inerente all'oggetto che uscirà dal Gachapon desiderato. Ovviamente parliamo di una palletta desiderata e non scontata, infatti in ogni macchinetta esiste una serie ben precisa di varianti dell'oggetto che viene distribuito. Esempio semplice: essendo musicista sono stato inizialmente catturato da un Gachapon che distribuiva miniature di strumenti musicali, il set di questi strumenti era composto da pianoforte, tromba, clarinetto, sassofono, etc...
Addirittura c'è una serie di pianoforti della Kawai per completare tutta la serie del solo strumento del pianoforte. Figuriamoci quanti modelli potrebbero esistere per ciascun strumento musicale dentro quella macchinetta! Questo pianoforte è realmente fedele, addirittura il coperchio superiore è apribile e c'è una graziosa asta sottilissima per bloccarlo in apertura. Strabiliante.
Altre miniature erano dedicate alle varie mascotte dei loro vari canali televisivi, di determinati articoli di bevande e cibo, personaggi famosi, animali rari (Balaeniceps rex!) e d'affezione, utensili della ferramenta e della cucina, oggettistica inerente alla metropolitana di Tokyo ed altre metropolitane e ditte, con le varie fermate scritte in mini targhette di plastica, oppure miniature di oggetti classici presenti nella vita quotidiana, etc... non c'è limite alla fantasia della collezione di tali miniature! Spesso anche fantasie estreme e "fantasie adulte", ve ne parlerò nei prossimi articoli! Una serie alla quale noi "boomer" non riusciamo proprio a sottrarci quando vediamo i Gachapon in Giappone è quella dei Kinnikuman e Wrestler vari, questo perchè in passato in Occidente erano approdati i pupazzetti di plastica dei "m.u.s.c.l.e.". In Italia per esempio abbiamo vissuto la saga dei famosi "Exogini", una linea lanciata nel 1987 dalla GIG. L'idea originaria di questi pupazzetti in realtà nasce in Giappone con delle plastiche morbide, simili alla gomma da cancellare, dell'eroe Kinnikuman, nel 1979.
Gommine subito reputate oggetto da collezione in terra nipponica. In Italia gli Exogini erano sbarcati con una storia sbrigativa e poco definita, pupazzetti alieni giunti sulla Terra in qualità di lottatori, fine! Erano di plastica rigida, sviluppati in quattro serie, in varie colorazioni e trasparenze.
I quaranta personaggi che costituiscono la prima serie erano chiamati Exogini in Italia, invece si chiamavano Muscleman in Francia e Musculman in Spagna. In Italia non erano presenti dentro i distributori di pallette, erano venduti in diversi formati unicamente dentro negozi fisici, per esempio in edicola. Erano distribuiti singolarmente in bustine, al prezzo di 500 lire italiane. In seguito furono venduti anche in barattoli da dieci pezzi (4500 lire) e piramidi da venti o quaranta pezzi (rispettivamente 8000 e 12000 lire). Si suddividono, oltre che per personaggio, anche per colore: i classici rosa, verde, arancione trasparenti e color carne opaco, più due colori speciali trasparenti contenenti brillantini. In seguito, visto l'enorme successo, furono ristampati in nuovi colori: blu metallizzato, grigio metallizzato, bianco, nero e viola e altre colorazioni. Un successo strepitoso! Ad oggi vengono collezionati e addirittura riprodotti con nuove varianti da numerose persone in diversi paesi dell'Europa. Ben quattro longeve serie di Exogini! Nulla in confronto alla longevità di questi pupazzetti in Giappone. Presenti saldamente ancora oggi, in numerosi formati, colorazioni e finiture.
Addirittura non solo gomma. Siccome il discorso Kinnikuman Vs Exogini è realmente titanico, vi ho allegato una foto della bellissima miniatura del pianoforte e di un bellissimo pezzo della "Bandai" di metallo. Così posso accendervi la curiosità per questi Gachapon e posso rimandarvi al prossimo articolo che si intitolerà "Gachagacha... pon!".
Alla prossima lettura, in questo viaggio nel passato ormai ritornato attuale, con ottime prospettive per il futuro. Praticamente lo specchio della loro filosofia.
A presto!

NEW GAME

PANG+

Anno: 2024
Editore.: Syx, Red Angel
Genere: Action Arcade
Piattaforma: Amstrad GX4000
Sito web: https://www.mojontwins.com/cepeceros/descargas/pang-plus-correccion-de-errores-y-mejoras-graficas-por-syx-y-redangel/

Nel lontano 1989 la Mitchell Corp. pubblicò un gioco arcade semplice ma dal gameplay avvincente. Il classico mangia gettoni che attirava un quantitativo enorme di giocatori.

Questo titolo era Pang, successivamente convertito su una moltitudine di sistemi dalla Ocean.

Lo abbiamo visto su C64, ZX Spectrum, Amstrad CPC, Amiga, MS-DOS e Atari ST.

Il gioco era piuttosto semplice nelle meccaniche: Il protagonista (o i protagonisti se si giocava in doppio) doveva distruggere degli enormi palloni rimbalzanti che infestavano i livelli.

Gli sferici oggetti una volta colpiti si dividevano in pezzi più piccoli fino a scomparire dallo schermo. Una volta eliminati tutti i palloni si passava al livello successivo.

Durante il gioco era possibile raccogliere potenziamenti o incappare in ostacoli.

Tra le versioni convertite una delle migliori era proprio quella uscita per la sfortunata console di Amstrad, il GX4000.

Una versione molto curata e quasi simile alla controparte arcade.

Nel 2024 gli sviluppatori Syx e Red Angel ci regalano questo bellissimo

Pang +. Grafica ottimizzata, miglior uso dei colori e un'azione più veloce.

Il tutto condito dal bel gameplay del titolo originale e dalla buona colonna sonora.

Un gioco da rigiocare e da apprezzare. Corrette a scaricarlo.

Questa versione “Plus” si può accostare senza dubbio alle versioni a 16 bit e, in termini di ottimizzazione, superarle.

di **Carlo Nithaiah Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 90%

Pang è un classico del suo genere e questa versione è praticamente perfetta.

» Longevità 90%

Valgono le dinamiche del gioco arcade. Pang è un gioco che va perfezionato partita dopo partita.

NEW GAME

SHADOW GANGS

Anno: 2022
Editore/Sviluppatore: Wave Game Studios, JKM Corp
Genere: Beat em up
Piattaforma: Sega DREAMCAST

Shadow Gangs, sviluppato da JKM Corp con sede nell'Isola di Wight e pubblicato dai sempre più prolifici Wave Game Studios, è stato a lungo atteso dal pubblico di appassionati.
Il porting per Dreamcast, forte di un finanziamento corposo di ben 25000 sterline ha raggiunto gli "scaffali" virtuali nel marzo del 2022.
Il titolo è una lettera d'amore a Shinobi. Un picchiaduro a scorrimento orizzontale con elementi platform che ricordano il favoloso titolo di Sega.
Lo scopo del gioco è quello di guidare il guerriero ninja DAN e mantenere l'ordine e la pace del mondo a suon di schiaffazzi e shuriken.
I livelli da affrontare sono 10 e sono piuttosto vari e pieni di nemici da combattere, ostaggi da liberare e oggetti da recuperare.
L'arsenale del nostro eroe si arricchisce di potenti attacchi ninja (che ricordano gli attacchi speciali di Streets of Rage) e di potenziamenti che migliorano la forza della nostra arma da lancio.
Il titolo gira perfettamente sul Dreamcast con i sui bei 60 fotogrammi al secondo. Fluido nelle animazioni e nello scrolling e con un bel design grafico per gli sprite e i livelli di gioco.
Shadow Gangs è ben rifinito anche per quanto riguarda le sue funzionalità audio. La musica è ben composta e si adatta perfettamente alle situazioni di gioco e gli effetti sonori sono ben realizzati.
Si tratta di una conversione diretta dalla versione PC e devo ammettere che si porta dietro gli stessi problemi ovvero la difficoltà estrema.
Un titolo difficile da giocare e con una curva non sempre bilanciata che vira decisamente verso l'alto.

Bisogna padroneggiare alla perfezione i comandi, affinare i riflessi e, soprattutto, memorizzare il flusso dei nemici livello per livello.
Quelli di noi le cui capacità sono state indebolite da giochi complessivamente più indulgenti usciti negli ultimi anni

GIUDIZIO FINALE

» Giocabilità 75%
Controlli precisi ma indice di difficoltà mostruoso già dal primo livello. Omaggia Shinobi alla perfezione.

» Longevità 70%
Adatto agli appassionati del genere.

si tengano alla larga dal titolo.
Bastava inserire una modalità "facile" che permettesse di padroneggiare al meglio il titolo e che, magari, rendesse il gioco più accessibile il titolo a chi non vuole passare un pomeriggio ad imprecare.
Allo stato attuale la modalità "Ninja nascente" soddisfa in qualche modo questa mia richiesta, ma solo perché dotata di 99 CONTINUE (in contrapposizione ai 15 della modalità classica).
Tuttavia, nonostante la difficoltà decisamente folle, Shadow Gangs è un titolo da provare.
Il gioco in versione fisica lo potete acquistare per 29,99 sterline e la versione digitale per 19,99.
Un titolo adatto ai veri hardcore gamers.

di **Giampaolo Moraschi**

NEW GAME

GREAT GIANA SISTERS

Anno: 2024
Editore/Sviluppatore: TCFS, Unreal, Csabo
Genere: Platform
Piattaforma: Plus 4
Sito web: https://plus4world.powweb.com/software/Giana_Sisters

Tutti conoscono Great Giana Sisters, in particolare i possessori di Amiga e Commodore 64.

Vuoi per la somiglianza a Super Mario e per una presunta causa con Nintendo, vuoi per la splendida colonna sonora di Chris Huelsback, vuoi per un buon gameplay e un bel design dei livelli... insomma è impossibile non conoscere questo titolo e risulta ancora più difficile non averci giocato mai una volta.

Un classico platform 2d con scrolling laterale con protagoniste due sorelle alle prese con cristalli da raccogliere, mostri da blastare e oggetti magici che le trasformavano da "bambine paffute" a punkettone arrabbiate.

Ben 33 livelli di gioco frenetico e tanto da esplorare, vedere, saltare e schiacciare. Un successo immediato che negli anni ha generato una grande fanbase.

Fu rilasciato originariamente sul 64 nel 1987 e l'anno dopo su Amiga, CPC e Atari ST. Programmato da Armin Gessert, con la grafica di Manfred Trenz e la colonna sonora del già citato guru del Sid Chris Huelsbeck sotto etichetta Time Warp.

Nel 2014, dopo seguiti, cloni, riedizioni e remake eccoci a parlare ancora una volta del titolo originale e lo facciamo su Plus 4, l'home computer meno compreso dalla stessa Commodore che sta vivendo una seconda giovinezza grazie alla scena di sviluppatori magiari che negli ultimi 2 anni hanno sfornato titoli nuovi e conversioni impensabili di una qualità sbalorditiva.

Giana è su Plus 4 ed è... bellissima!

Una conversione perfetta. Ci sono tutti i livelli, tutti i mostri e tutta la follia presente nel titolo del C64.

La grafica si muove molto bene e non risente assolutamente di rallentamenti eccessivi o di scrolling poco fluido. Appare più colorata e gradevole grazie alla tavolozza del Plus 4.

L'audio riproduce bene quello sentito su SID ma coi limiti dei canali sonori del TED.

GIUDIZIO FINALE

» Giocabilità 90%
C'è tutto il colorato mondo di Giana Sisters e si muove davvero con stile anche su Plus 4.

» Longevità 95%
33 livelli di gioco e un grandissimo gameplay.

Un lavoro perfetto del team capitanato dallo sviluppatore indipendente TCFS e del suo team composto da Unreal alla grafica e Csabo alle musiche.
Recuperatelo e giocateci perché merita anche in questa sua nuova/vecchia veste su Plus/4.
Clamoroso!

di **Carlo Nithaiah Del Mar Pirazzini**

KING OF DEMONS

Anno: 1995/2024
Editore/Sviluppatore:
Genere: Action/Platform
Piattaforma: Super Nintendo
Sito Web: http://retro-bit.com/majyuo/

Il 1995 del Super Famicom era un focolaio di grandissimi titoli. Giochi di ogni genere e anche diversi prodotti con ambientazioni orrorifiche.
Majyuo (titolo originale del titolo) fu uno degli appartenenti di questa ondata di titoli splatter. Un titolo mai localizzato in lingua inglese almeno fino ad oggi.
In occasione della sua uscita grazie a Retro-Bit, parliamo di uno degli action game più adulti per il Super Nintendo.
Pensate, per anni il 16 bit di Nintendo è stato considerato una macchina con titoli per famiglie o quasi. Titoli solari e ottimisti, grandi rpg e giochi d'azione. Eppure nella sua vastissima ludoteca è possibile reperire alcune gemme horror destinate ad un pubblico maturo.
Questo King of Demons è uno di questi. La KSS ci ha donato un action game crudo e dalla trama forte.
Abel è il protagonista dell'avventura. Un giovane eroe tradito dal suo migliore amico che ha venduto la sua anima ai demoni. Il mascalzone ha sacrificato la povera moglie del nostro eroe, la figlia e persino il piccolo gatto per poter risvegliare Lucifero, il re dei demoni, dal suo sonno eterno.
Abel decide di vendicarsi ma fallisce. Il gioco inizia con la morte di Abel per mano di un demoniaco Bayer, ma lo spirito della moglie uccisa lo riporta in vita e a questo punto comincia davvero l'avventura.
L'inferno è un luogo vivace, rappresentato da sette livelli tanto vari quanto dettagliati.

Gli sfondi ad esempio sono semplicemente mozzafiato.
L'eroe si muove in modo soddisfacente lungo le piattaforme con le sue abilità di doppio salto e di scivolamento (fondamentale quest'ultima).
La sola forma umana del nostro eroe però non basta contro i demoni; in soccorso possiamo utilizzare la capacità di mutarsi dopo la sfida con ogni boss di fine livello. Una volta

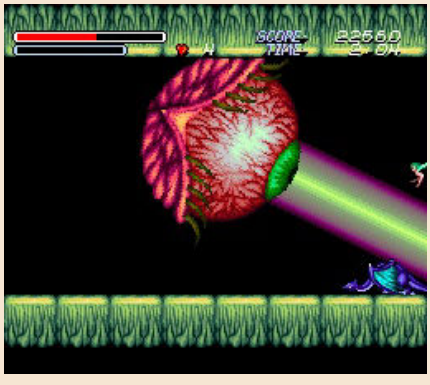
mutato Abel diventa uno strano costrutto che ricorda Devilman.
Interessante questo plus in gioco che permette di affrontare alcuni livelli davvero impossibili in forma umana.
Vale la pena soffermarsi che questo è un gioco dal tono molto oscuro. Non c'è gioia, né emozione, c'è tanto gore e la trama è molto malinconica.
Come già accennato il comparto tecnico è decisamente di grande livello ma il gioco soffre di una bassa longevità. Il sette gironi infernali sono pochi e in circa mezz'ora di pratica si porta a casa la pellaccia.
Una menzione speciale va alla colonna sonora che contiene alcuni brani memorabili realizzati da Hiroshi Lizuka e Tomohiro Endo.
Rendono l'inferno un luogo ancora più inquietante.
Il titolo negli anni è diventato un pezzo di culto e finalmente è possibile giocarlo in versione fisica dopo aver apprezzato la traduzione del 2005 in una interessante patch.
Un titolo che però merita di essere riscoperto e magari aggiunto in formato fisico nella propria ludoteca.

di **Roberto Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 90%
Ottimo sistema di controllo e un bel level design. Il gioco si lascia esplorare molto bene.

» Longevità 75%
I livelli sono troppo pochi. Ne avrei voluti giocare ancora e ancora…



GO FOR IT! GOEMON
THE GRAND CYCLE OF OEDO

Anno: 2001
Editore/Sviluppatore: Konami / Accadiez
Genere: Platform/Adventure
Piattaforma: Playstation 1
Patch uk: https://www.romhacking.net/translations/7256/

Così Giapponese e così impossibile da non amare! Questa frase riassume perfettamente ciò che penso della serie Ganbare Goemon.

Un serie nata su MSX e NES e successivamente portata alla ribalta su Super Famicom con 4 capitoli eccellenti.

Oggi è il turno di parlare di questo suo capitolo per la prima Playstation che recentemente ha ricevuto una succosa patch per esser giocato in lingua inglese.

Il gioco parla del terribile malvagio Ecorori che rapisce la bella Omitsu. Tocca quindi al prode Goemon e al suo gruppo di squinternati amici (il ninja obeso con tendenze bisex Ebisumaro, il robot punk Sasuke e la bella Yae) salvarla.

Il gioco si ispira tantissimo al secondo capitolo visto su Super Nintendo che abbiamo recensito sul numero 18 di RMW italiano. Avremo quindi alcuni livelli di puro platform a scorrimento orizzontale, accompagnati da boss giganteschi in prima persona, seguiti da livelli esplorativi in città e villaggi. Sono presenti apparizioni dei folli personaggio di tutta la saga e di alcune facce viste sul capitolo per Nintendo 64, il tutto condito da una bella dose umoristica onnipresente nella serie. Rispetto ai capitoli sul Nintendo a 16 bit, qui si fa uso di una grafica pre renderizzata che al tempo svolgeva il suo lavoro. Al giorno d'oggi appare piuttosto strana e, come tante altre produzioni 3d dell'epoca, non è invecchiata benissimo.

Ad ogni modo il titolo risulta colorato

e pieno zeppo di oggetti, mostri e piattaforme in ogni dove.

Particolarmente apprezzabile la colonna sonora tratta dal primo capitolo del titolo e da Monoke

Suguroku (il tiolo per Nintendo 64).
Se dovessi trovare una definizione per questa versione PS1 dire che si tratta di un “best of” dei migliori elementi della saga condito dal proverbiale gameplay e dal bel level design in puro stile Konami.
La traduzione effettuata da Accadiez rende più godibile il gioco e si applica al titolo (tramite patch della rom, a patto di possedere un originale) attraverso il classico xdelda, applicazione leggerissima scaricabile gratuitamente dalla rete.

L’ostacolo della lingua nipponica non è mai stato bloccante nella versione originale, ma leggere il testo in inglese rende più scorrevole la trama e facilita le risate.
Un titolo da riscoprire per la vostra PS1 e da provare anche in questa sua versione nella lingua d’Albione.

di **Roberto Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 90%
Level design bello e un grandissimo lavoro in termini di game play in puro stile Capcom che, grazie alla patch, possiamo approcciare ancora meglio.

» Longevità 80%
E’ uno dei titoli che si terminano più velocemente ma è un gran bel giocare.

NEW GAME

ALICE SISTERS

Anno: 2024
Editore/Sviluppatore: PPixel Heart/Orion Soft
Genere: Platform/Puzzle
Piattaforma: Atari Falcon
Sito web: https://orionsoft.itch.io/alice-sisters

Nel dicembre del 2022 la nostra cara Ingrid aveva già recensito Alice Sisters per il Sega Megadrive appioppandogli 90% e 95% in giocabilità e longevità.

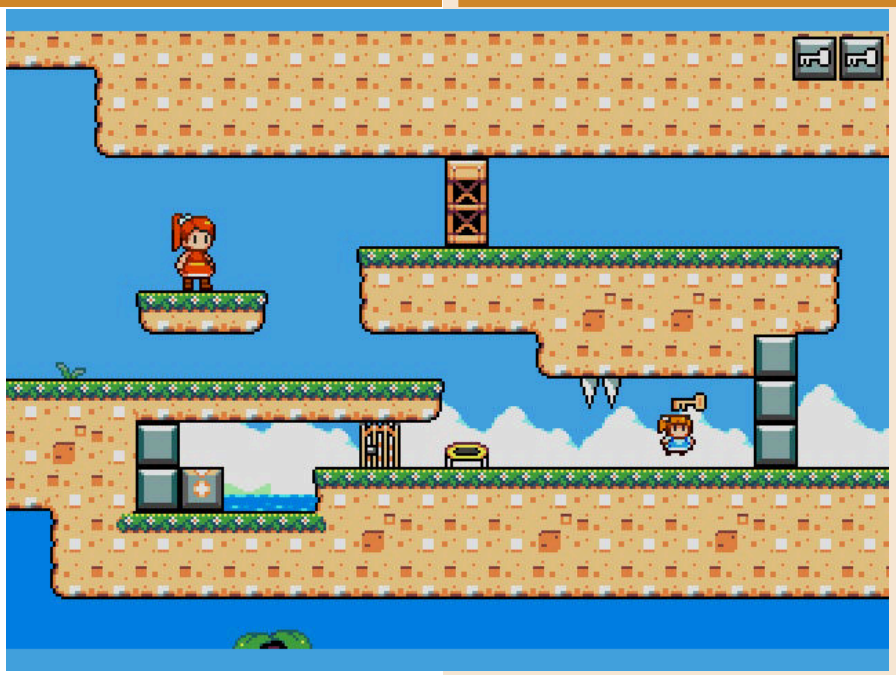
Era entusiasta del titolo per la modalità di gioco e per i tanti livelli presenti.

Dopo due annetti ecco arrivare la versione per Atari Falcon, sfortunato personal computer sviluppato da Atari Corp nel 1992. Una piattaforma poco conosciuta ma dalle caratteristiche particolari, ammazzata dal costo esagerato e con una strana interfaccia a 32bit con un bus a 16 che garantiva la compatibilità con il software ST, ma tarpava un po' le ali sul "futuro".

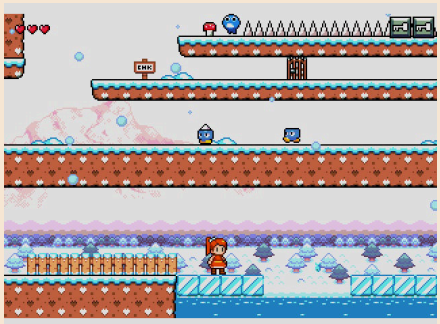
Ma bando alle ciance e passiamo al gioco.

Un malvagio orco ha catturato la mamma di due tenere e paffutte bambine. Il nostro compito è aiutarle lungo 28 livelli distribuiti su 4 mondi di gioco a salvare la "genitrice 2" riportandola a casa.

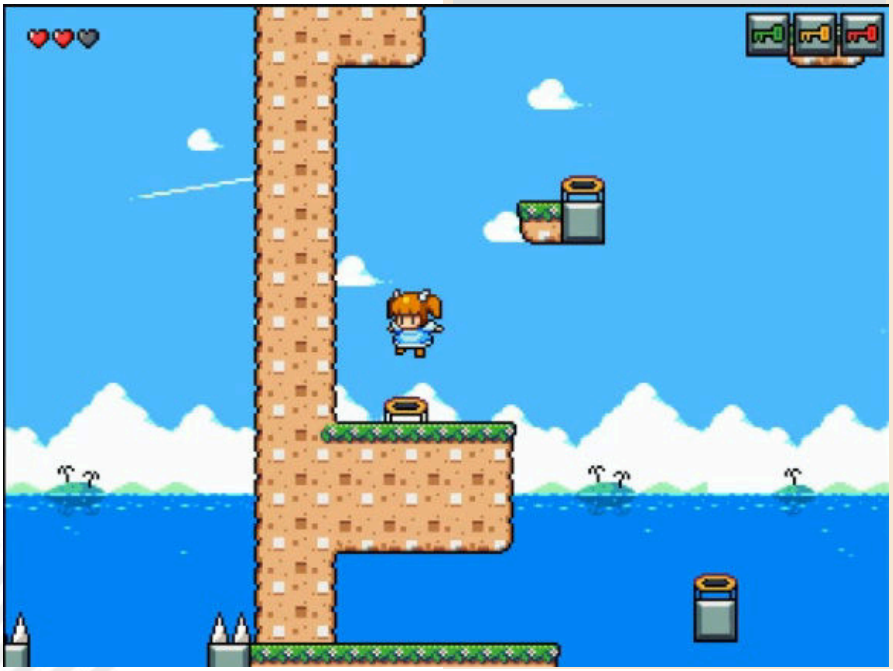
Saremo alla guida di entrambe le bambine che si "Switchano" premendo in basso il joystick. La più piccola e paffuta può cambiare dimensione attraverso funghi magici (non pensate male) e saltare più in profondità; mentre la più grande può difendersi lanciando una sfera rossa contro i nemici.

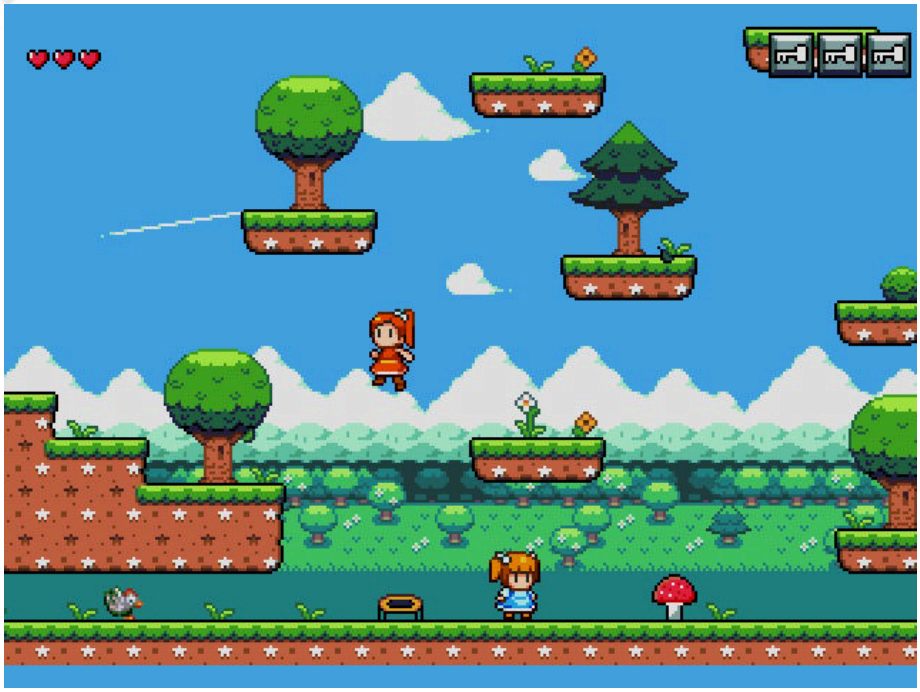


Lo switch mette in moto delle combinazioni essenziali per poter terminare il livello e proseguire nel successivo.



Come per la versione MD, anche quella su FALCON è bella da vedere e da





GIUDIZIO FINALE

» Giocabilità 85%
L'uso del joystick spiazza inizialmente (il salto si fa premendo in alto e non come su megadrive col secondo tasto) ma il gioco è ben sviluppato e divertente.

» Longevità 95%
Tanti livelli, 4 modalità di gioco e diverse ore da passarci sopra.

sentire. La grafica è gradevole e molto colorata. I personaggi sono ben caratterizzati e si muovono bene.

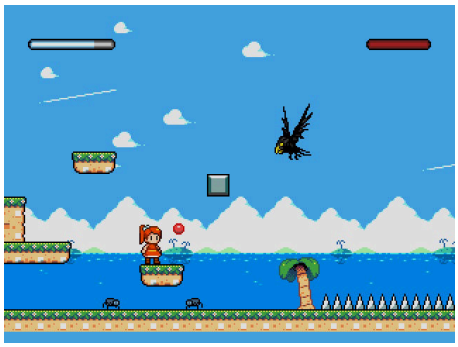
È un titolo divertente e che tiene incollati allo schermo grazie ad una difficoltà ben bilanciata e ad uno studio dei livelli ben fatto.

Questa versione FALCON è a pagamento e costa 12 dollari. Nel file che verrà acquistato troverete anche la versione PC (identica).

Il gioco richiede 8MB di RAM e l'installazione su disco fisso. È anche compatibile con l'emulatore Hatari.

Non mi resta che concludere confermando la validità del gioco e consigliandovi l'acquisto e il supporto ad OrionSoft.

di **Giampaolo Moraschi**



NEW GAME

TETRIS

Anno: 2024
Editore/Sviluppatore: Eric Carr
Genere: Puzzle
Piattaforma: Atari 8bit (800 48k, XE/XL)
Sito web: https://bunsen.itch.io/tetris-atari-8-bit-by-eric-carr

Il concorso annuale Basic 10Liner celebra nel 2024 la sua 13° edizione. Questa competition per sviluppatori è incentrata su brevi linee di Basic, aperta per tutti i computer a 8bit.

L'edizione 2024 scade a fine marzo 2024.

Si può partecipare a più categorie:

PUR 80 – Gioco su 10 righe max 80 caratteri per riga logica.

PUR 120 – Gioco su 10 righe max 120 caratteri per riga logica.

EXTREM 256 – Gioco su 10 righe max 256 caratteri per riga logica.

SCHAU: Gico su 10 righe max 256 caratteri per riga logica.

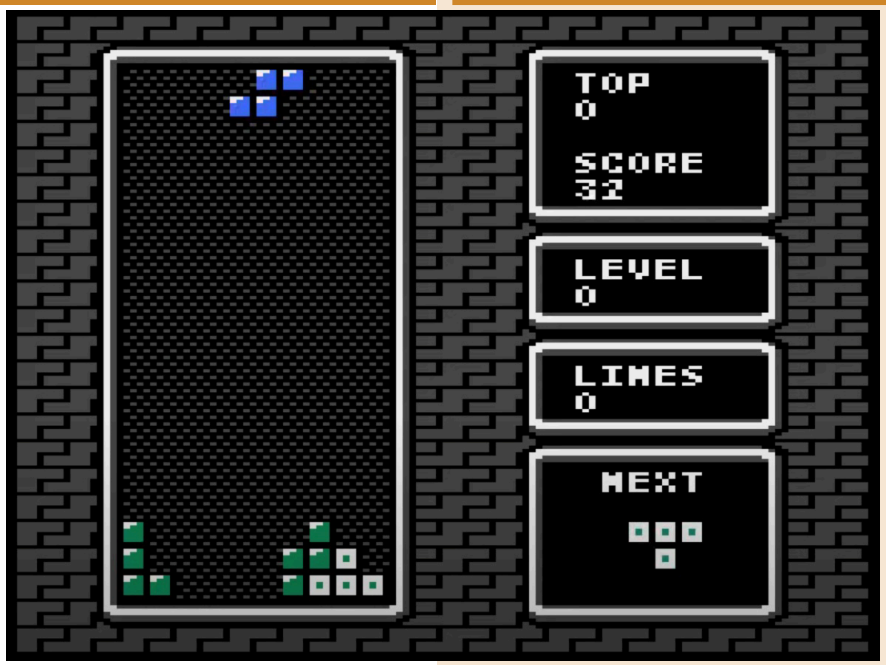
Le prime tre categorie sono esclusivamente per giochi Basic, la SCHAU invece può essere qualsiasi cosa: una demo, una utility o un programma applicativo.

I programmi inviati che non rientrano nelle categorie vengono raggruppati nella categoria WILD.

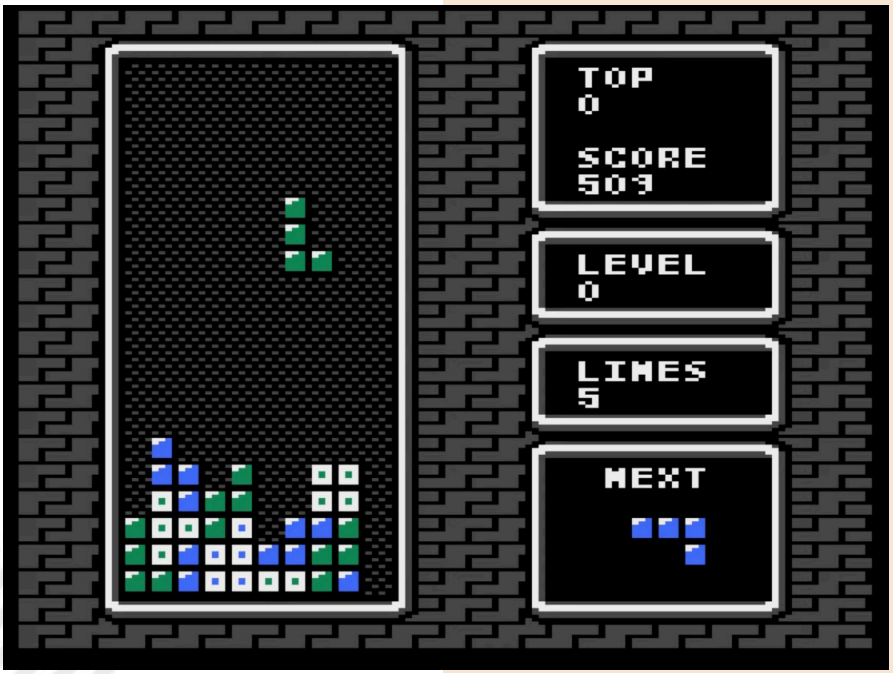
Anche Retromagazine in passato ha partecipato alla competition con il suo redattore e sinclairista Alberto Apostolo (ci manchi in redazione Albe!!!).

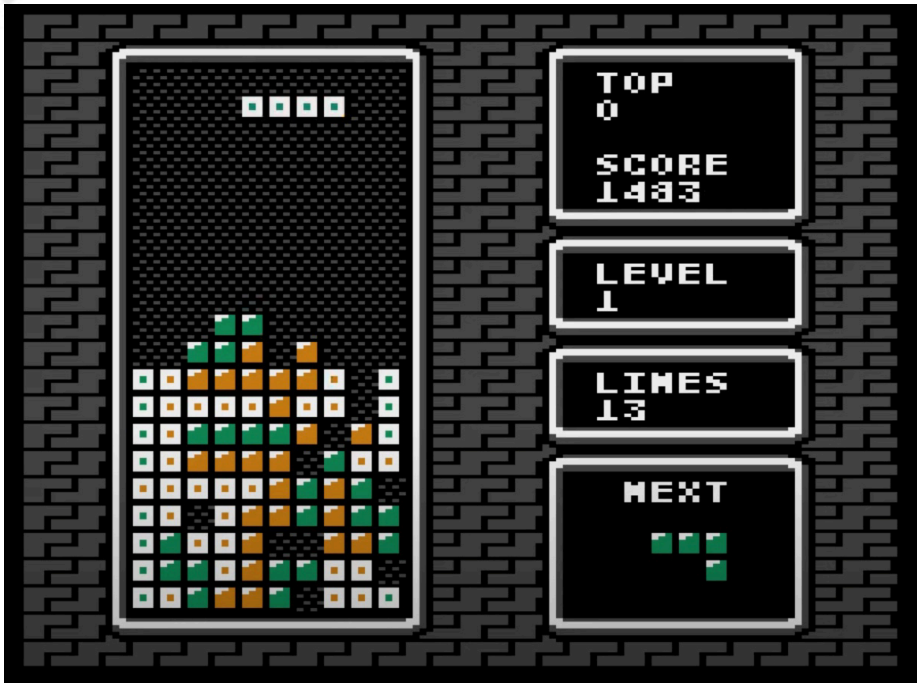
È una manifestazione che personalmente adoro e che mette in mostra sempre l'ingegno dei tanti partecipanti.

Tetris di Eric Carr è l'esempio di come fare le cose per bene in poche righe di codice.



Il gioco non ha bisogno di spiegazioni o presentazioni. È Tetris e quindi, se non avete vissuto in un vulcano negli ultimi 40 anni, non mi dilungherò nelle spiegazioni.





GIUDIZIO FINALE

» Giocabilità 90%
Possono 10 righe essere perfette? Si. Qui c'è tutta la magia del TETRIS ed un grande omaggio alla versione NES.

» Longevità 99%
Eterno.

Quello che posso sicuramente dire è che è una versione davvero ben fatta. Realizzata ispirandosi a quella per Nintendo NES.

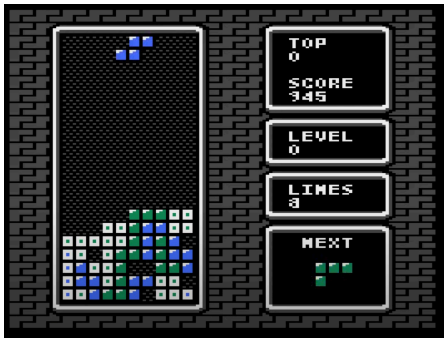
È stato utilizzato un set di caratteri personalizzati, un codice musicale da zero, un compilatore FastBasic 4.6 per compilare in XEX, Visual Studio per il codice e tanta passione.

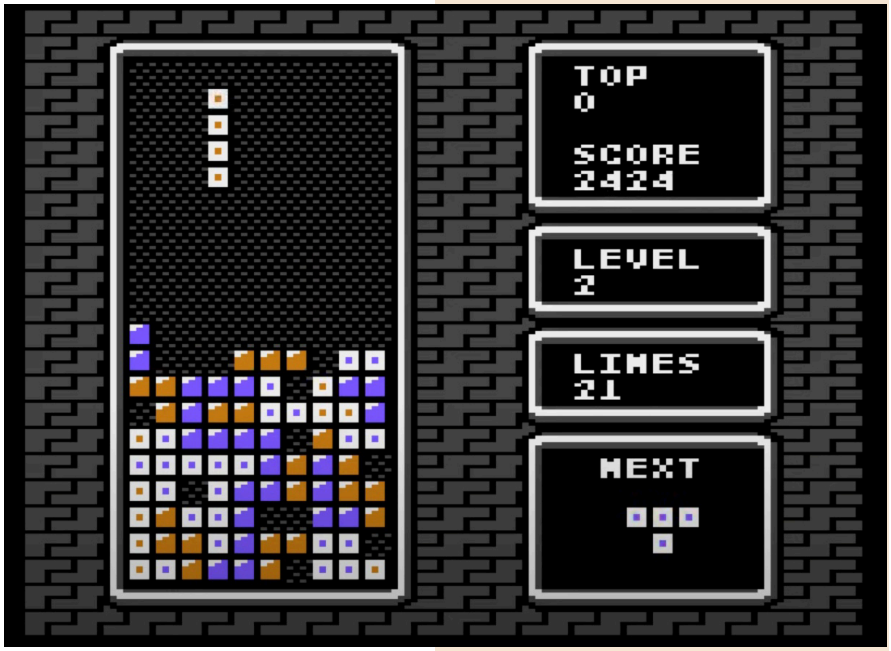
Un grande Tetris, forse uno dei migliori a cui abbia giocato da anni!

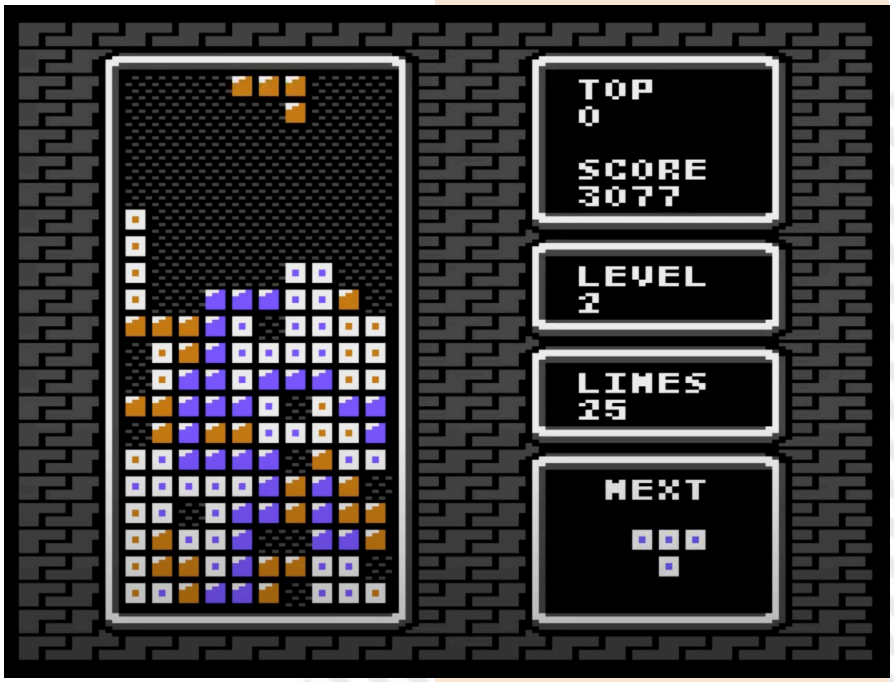
Veloce, divertente e praticamente millenario.

L'ennesima dimostrazione di una comunità, quella dei retro sviluppatori, viva e capace di fare cose impossibili... in sole 10 righe!

di **Carlo Nithaiah Del Mar Pirazzini**







NEW GAME

XIAOMEI AND THE FLAME DRAGON'S FIST

Anno: 2023
Editore/Sviluppatore: PiXEL
Genere: Beat em up
Piattaforma: Sega Megadrive
Sito web: https://leoful.com/xiaomei-and-the-flame-dragons-fist-en/

Sviluppato dai designer giapponesi PiXEL , Xiaomei and the Flame Dragon's Fist è un omaggio ai vecchi giochi di combattimento a scorrimento laterale a 8 e 16 bit.

C'è un po' di Kung Fu Master in questo titolo, ma poco…. poco poco.

L'azione si svolge su livelli a scorrimento laterale in cui orde di ninja e spadaccini corrono continuamente verso Xiaomei. La maggior parte richiede uno o due colpi per essere sconfitta dall'occasionale mini-boss che avrà tecniche di attacco più avanzate. Ogni livello termina con un boss complicato che richiederà abilità per comprendere come farlo fuori.

Si può colpire con calci e pugni ed è possibile attaccare saltando premendo verso l'alto e combinando con il tasto del colpo preferito.

Mentre attacchiamo il nemico, un misuratore speciale si accumula consentendo di lanciare una potente palla di fuoco capace di far piazza pulita dei nemici sullo schermo.

Xiaomei è un gioco difficile per impostazione anche in modalità easy. Rende sicuramente omaggio ai gloriosi giochi del passato ed hai kung fu movie, ma lo fa con una pochezza tecnica imbarazzante.

Animazioni scarse, un pessimo uso del colore e pochissima varietà negli avversari e nei livelli. C'è poco fascino nel personaggio della protagonista e alcune scelte in merito alle collisioni tra sprite sono, come dire... particolari! Vogliamo parlare del reparto audio? Anche qui c'è da piangere. Colonna sonora poco incisiva e terrificanti effetti sonori.

Su tutto questo un gameplay legnoso. L'uso sbagliato dei tasti del Megadrive (uno completamente non operativo) si mescola alla banalità dei livelli di gioco e alla difficoltà decisamente

sbagliata.
Come dice un caro amico... "sembra Kung Fu Master, ma più bello!" (ovviamente seguito da una risata sonora).
Di picchiaduro a scorrimento il Megadrive ne ha tantissimi e quasi tutti di livello alto o quantomeno dignitoso. Questo lasciatelo perdere. Inoltre la cart costa decisamente tanto.
Lasciatelo lì.
P.s il titolo è disponibile anche per le nuove piattaforme...
Dio ce ne scampi!

di **Carlo Nithaiah Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 30%
Il comparto audio/visivo è scarso, ma il vero punto dolente è un gameplay terribile e una pochezza di idee assurda.

» Longevità 10%
Mi sono divertito quando lo schermo era ancora spento e la console... pure.

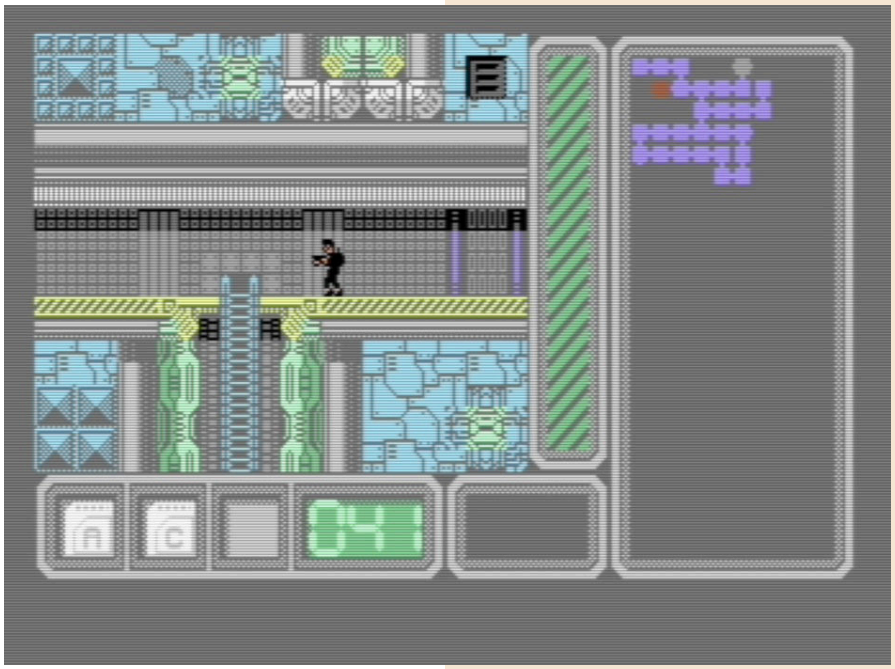
NEW GAME

HARHARGON

Anno: 2024
Editore/Sviluppatore: Icon 64 – Psyktronik
Genere: Action/Platform
Piattaforma: Commodore 64
Sito web: https://www.psytronik.net/newsite/index.php/c64/147-harharagon

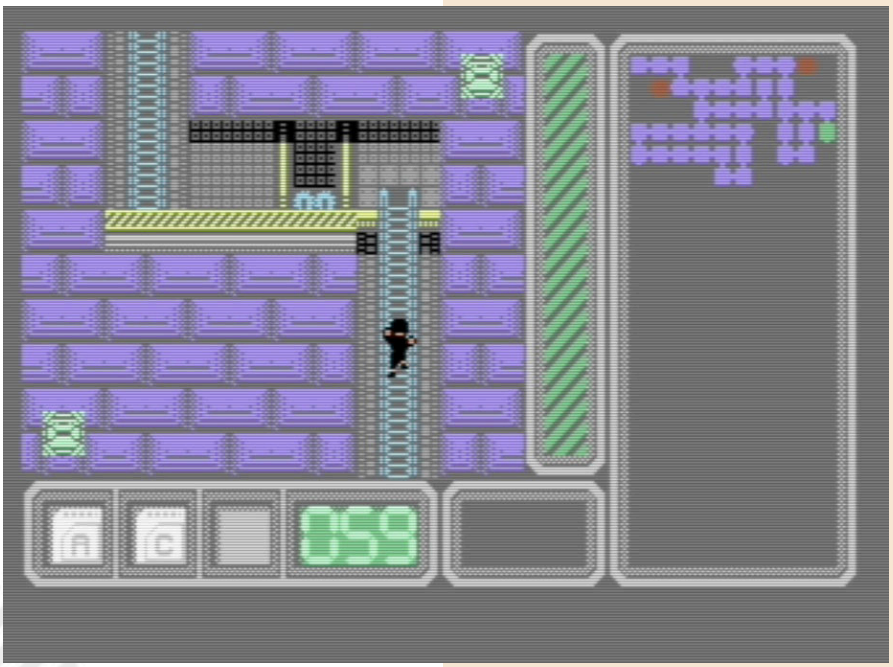
Un gioco dalle tinte spy, che ci ha ricordato molto Impossible Mission per struttura e realizzazione e che si è presentato come i "Budget game" di una volta.

Harharagon ci mette nei panni di un prigioniero in un carcere di massima sicurezza. Siamo stati incriminati per un crimine che non abbiamo mai commesso, solo perché abbiamo avuto la relazione con la figlia della persona sbagliata al momento sbagliato.
Ci troviamo all'interno di una prigione/labirinto e il nostro scopo sarà quello di esplorarla, recuperare 18 pezzi della capsula di salvataggio e fuggire. La prigione conta 230 stanze e non sarà un compito semplice esplorarle tutte. Numerosi guardiani cybernetici si occuperanno di tutti coloro che tentano la fuga… con ogni mezzo necessario!



Ben sviluppato il meccanismo di fuoco ad 8 direzioni, che permette di colpire con efficacia gli avversari che ci piomberanno addosso senza pietà.
La musica di Saul Cross fornisce una singola melodia con forti influenze

Se non bastasse la semplice esplorazione, la ricerca dei pezzi mancanti e i robot nemici, il nostro eroe dovrà anche occuparsi del recupero di alcuni cristalli che permettono di aggiornare la potenza di fuoco dell'arma in dotazione. I cristalli vengono rilasciati dai nemici distrutti o recuperati nel percorso.
Icon64 ha realizzato un run'n gun molto solido che fornisce circa 90 minuti di gioco stile arcade.



Lo stile grafico ad alta risoluzione è funzionale e molto attraente e il personaggio si muove bene e con dettaglio nelle animazioni.

GIUDIZIO FINALE

» Giocabilità 80%
Un titolo ben congeniato nella sua struttura. Semplice nei comandi e con un buon sistema di fuoco del protagonista.

» Longevità 80%
Offre un buon grado di sfida con le sue 230 stanze. Alcune sono ripetute ma non ce ne accorgiamo nemmeno.

anni 80. Martellante e ben sviluppata che utilizza in modo completo il SID.

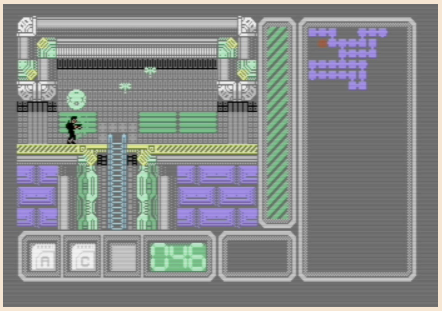
Il gioco è semplice da apprendere e fornisce un grado di sfida crescente. La schermata in game mostra l'azione in gioco con in basso il menù degli oggetti e lateralmente la mappa della prigione che, esplorata, ci mostrerà il percorso fatto, le uscite e le zone contenenti oggetti e quant'altro.

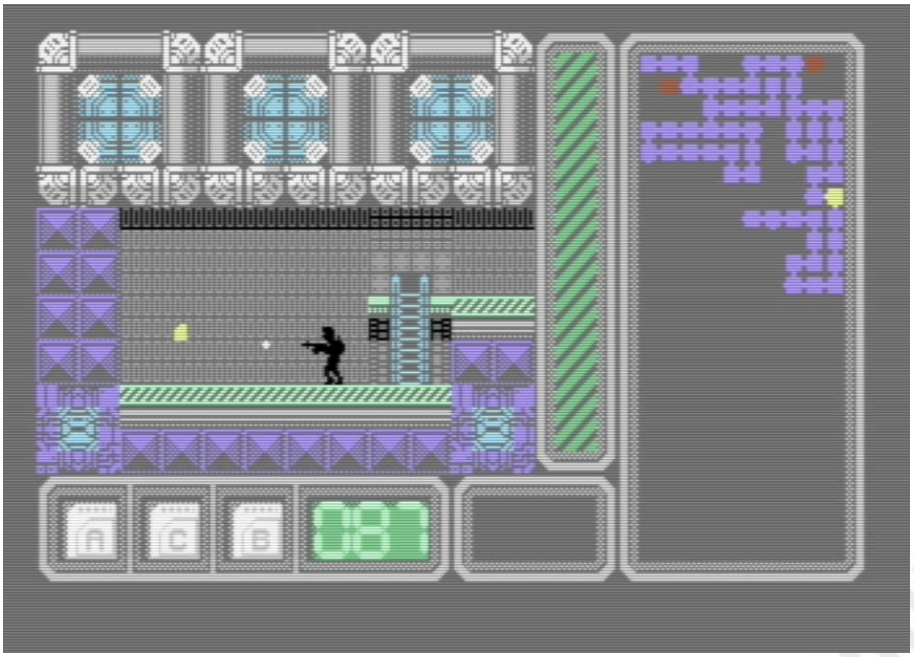
Se dobbiamo trovare il classico pelo nell'uovo possiamo affermare che alcune stanze si ripetono per schema e riproduzione dei nemici. Nulla di che in termini di giocabilità e di divertimento. C'è molto da apprezzare in Harharagon.

È ben realizzato nel suo insieme anche se non offre nulla di nuovo nel panorama di questo genere su C64.

Però è divertente e di sicuro vi piacerà

di **Carlo Nithaiah Del Mar Pirazzini**





NEW GAME

TRUCHO

Anno: 2024
Sviluppatore: TheNestruo
Genere: Action game
Piattaforma: MSX
Sito web: https://thenestruo.itch.io/trucho - https://msxcartridgeshop.com/

Bello! Bello! Bello! Questo Trucho è un titolo semplice ma curato sotto ogni punto di vista e anche divertente. Ispirato ai "giochi di pesca" per Pc o console, questo titolo ci mette nei panni di un pixeloso pescatore intento a partecipare ad una serie di competizioni.

A dire il vero i pescatori selezionabili sono tre, ognuno con le proprie abilità di tiro e tutti ben caratterizzati.

Una volta selezionato il nostro personaggio si viene catapultati nella zona di gara e parte l'azione.

Selezioneremo il tipo di esca in base alle nostre competenze e lanceremo la nostra esca facendo attenzione a ciò che accade nella parte bassa dello schermo di gioco.

In questo punto appariranno diversi pesci (dalle caratteristiche differenti) e starà a noi farli abboccare al nostro amo.

Impresa non facile tanto quanto quella di riportarli sulla barca.

Una volta recuperati i pesci prestabiliti per terminare il livello, si passerà al successivo e la difficoltà aumenterà in modo graduale.

Sulla carta semplice, ma vi assicuro che vi appassionerà tanto quanto uno dei moderni fishing game.

La pesca richiede pazienza e anche questo Trucho. Attendere il momento giusto e poi tirare le prede in barca richiede precisione per gestire la barra della forza con il proprio joypad/joystick... e dimenticavo di dirvi che è presente anche un tempo per portare a termine il livello.

Il lavoro fatto da TheNestruo è ben fatto e funziona alla grande. Mi sono piaciute moltissimo alcune scelte

cromatiche e la possibilità di utilizzare alcune tecniche in gioco come il richiamare l'attenzione dei pesci verso l'esca.

Un gioco sviluppato con criterio tecnico e dotato anche di una simpatica musichetta e dei buoni effetti sonori.

Certo è un gioco mirato agli appassionati di pesca o a chi cerca qualcosa senza troppe pretese, ma è sicuramente da provare.

Il titolo è disponibile a pagamento in due versioni: in versione fisica, corredata da un bel manuale, da una confezione davvero bella e un gadget (oltre alla cartuccia); e in versione flash rom a soli 5,26 con i file necessari per giocare.

di **Carlo Nithaiah Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 90%

Semplici ma completi comandi di gioco ed un gameplay davvero interessante e che tiene incollati alla sfida.

» Longevità 80%

Se amate questo tipo di giochi è perfetto. Se cercate qualcosa di più "dinamico" forse è il titolo meno indicato. Resta comunque un gioco davvero ben fatto e con una curva della difficoltà eccellente.

NEW GAME

GOOD KNIGHT

Anno: 2024
Editore/Sviluppatore: Psytronik Software - Icon64
Genere: Platform
Piattaforma: Commodore 64
Sito web: https://psytronik.itch.io/good-kniight

Ci risiamo… il solito drago sputafuoco che soffre di solitudine e con manie di grandezza ha rapito la principessa (di turno).
Il re è molto preoccupato e non sa a chi rivolgersi. Su Commodore 64 tutti gli eroi sono impegnati in missioni eroiche per salvare mondi, principesse e loro stessi e, al momento, nessuno è disponibile (Sir Arthur di Ghost 'n Goblins poi … non ne parliamo!).

Non resta che rivolgersi a Sir Darren di Durham, noto come il Buon Cavaliere. Sicuramente un buon ragazzo ma decisamente non il più eroico.
È un guerriero più adatto ai combattimenti a tavola contro primi, secondi e dessert che alla pugna contro mostri e creature bizzarre.
Ma la vita è dura e questo passa il "convento" si dice no?
Ad ogni modo Sir Darren accetta e l'avventura comincia.
Good Knight è platform game ben fatto e che ricorda Hunchback.

Sir Darren si muove da sinistra a destra in schermate singole. Ogni livello presenta una sfida di diverso tipo: salti di piattaforme, lanci di palle di fuoco, seghe rotanti, cavalieri nemici, mostri, frecce avvelenate e chi più ne ha più ne metta.
A suo vantaggio può saltare per evitare tutto questo, usare piattaforme, corde e la sua mobilità in generale.
Il tutto fino ad arrivare al drago malvagio che tiene in ostaggio la principessa.
Sono circa 40 livelli (con alcuni schermi bonus) da affrontare per una sfida che, una volta appresa con attenzione, richiederà una 30 di minuti.
La grafica è semplice ma curata e animata con cura. Bella anche la scelta cromatica utilizzata.

Colonna sonora in stile SID-medievaleggiante che ci accompagna per tutta la nostra "tenzone" ma totale assenza degli effetti sonori.
Good Knight è un gioco "bastardo"! Sembra semplice ma è letale. Se si sbaglia il salto… si muore! Se si prendono male i tempi di esecuzione delle armi avversarie… si muore! Se si affronta tutto frettolosamente e senza pensare… si muore!

GIUDIZIO FINALE

» Giocabilità 75%
Nella sua disarmante semplicità è un titolo ben strutturato e decisamente letale nel suo metodo di apprendimento dei livelli.

» Longevità 75%
Si muore spesso, molto spesso, ma la voglia di vedere tutti i livelli è tanta e non ci si stacca dal pad tanto facilmente.

Richiede riflessi pronti e memoria per ogni livello. Queste due cose sono fondamentali.
Questa letalità però non influisce sul divertimento, anzi aumenta la voglia di proseguire e di vedere cosa c'è più avanti.

Il titolo è disponibile sul portale itch.io degli sviluppatori al costo di 3,99 dollari. Prezzo budget ottimo.
Girà perfettamente in real, in emulazione e sulla THEC64 family.

Consiglio? Fornitevi di un buon pad di gioco… e di taaaaaanta pazienza!

di **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

SAVAGE PRINCESS 2

Anno: 2024
Editore/Sviluppatore: Monster's Legs
Genere: Platform
Piattaforma: ZX Spectrum
Sito web: https://monsters-legs.itch.io/savage-princess-2

L'umorismo non manca agli autori di questa saga.

La principessa selvaggia, dopo le fatiche della sua prima avventura, si reca al pub.

Qui da sfogo alla sua pantagruelica fame e alla sete più assoluta fino ad arrivare ad una (meritata) sbornia.

Mentre pisola saporosamente viene svegliata bruscamente da Eldritch e dalle urla proveniente dal fuori la locanda.

Chthulu si è risvegliato! Ed è incacchiato nero! Perché lo hanno svegliato dal sonno eterno e perché è un Grande Antico che mira alla conquista del mondo.

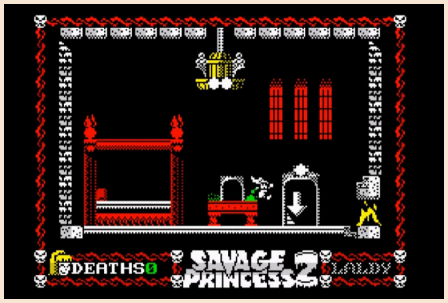
Ava, la principessa selvaggia, parte così per l'avventura con lo scopo di ricacciare il mostro tentacolare negli abissi della sua città R'lyeh… ovviamente dopo aver imprecato come uno scaricatore di porto…

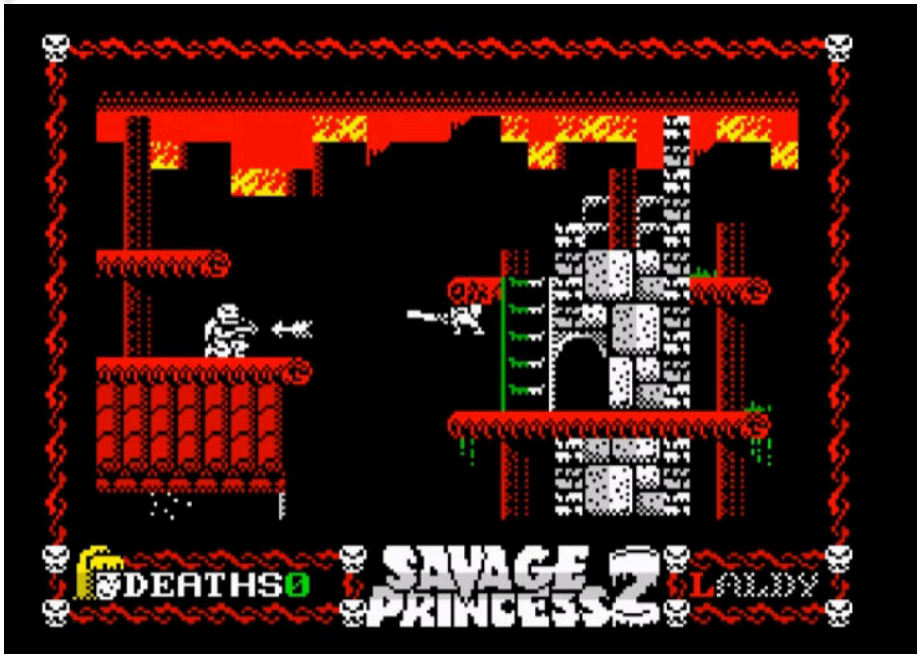
Ribadisco gli autori sono dei burloni oppure sono folli, ma in ogni caso Savage Princess 2 è divertente.

Un platform semplice a mono schermata animato con cura e dalla grafica "pucciosa" che si lascia domare senza problemi.

Semplice da apprendere e con una difficoltà graduale.

Per proseguire nell'avventura è necessario recuperare cinque gettoni: i primi 3 sono nel primo livello e 1 gettone nei livelli successivi. Solo recuperati queste monetone permetteranno a Ava di raggiungere il titanico Chthulu.





GIUDIZIO FINALE

» Giocabilità 85%
Il titolo è ben sviluppato sotto tutti i punti di vista e si lascia gustare. L'umorismo folle è il punto di forza.

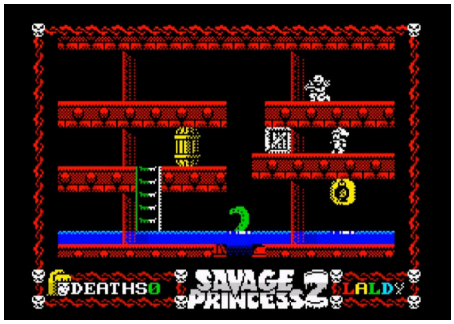
» Longevità 80%
Non è lunghissimo ma è lodevolissimo.

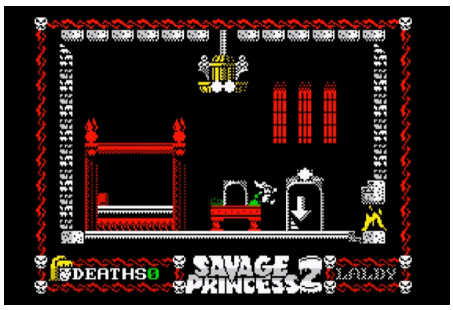
Ottimo uso del MPAGD e simpatica l'idea di essere rimproverati (insultati) ogni volta che si perde una vita.

Il file è .tap e può essere caricato su qualsiasi emulatore ZX Spectrum senza problemi oppure riprodotto su hardware reale.

Se avete voglia di farvi quattro risate e di affrontare un titolo divertente questo Savage Princess 2 fa al caso vostro.

di **Giampaolo Moraschi**





NEW GAME

GODZILLA BALLS

Anno: 2024
Editore/Sviluppatore: Paxanga Software
Genere: Puzzle/Action
Piattaforma: MSX
Sito web: https://paxanga.itch.io/godzilla-balls

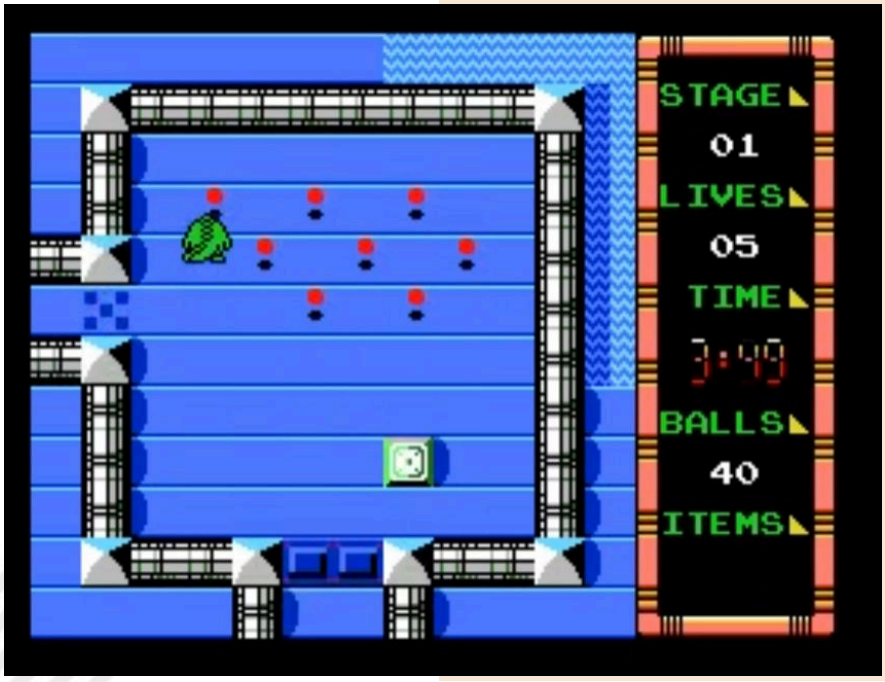
Il mostruoso Godzilla ha da poco compiuto 70 anni. Dalla sua prima apparizione nel film del 1954 è stato incoronato Re dei Mostri e da allora nessuno è più riuscito a spodestarlo. È stato anche uno dei primi Kaiju ad apparire nella fantascienza cinematografica nipponica. Il termine indica i mostri giganti nati a causa delle radiazioni dei test atomici e dei bombardamenti americani sul Giappone nel corso della seconda guerra mondiale.

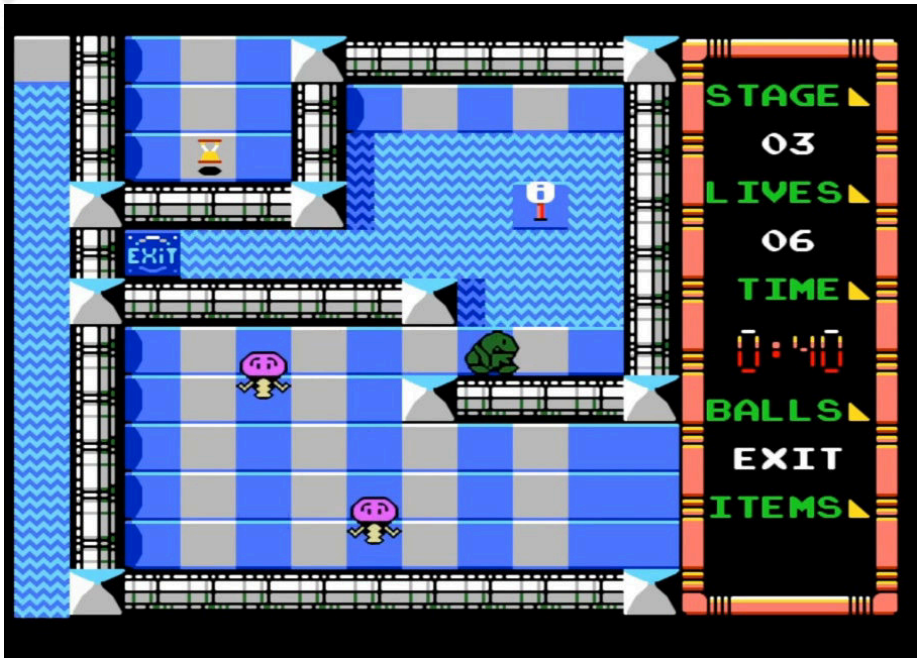
Il concetto che troviamo alla base di questa mitica creatura è semplice: l'umanità ha scatenato un potere che non può controllare e la sua presunzione l'ha condotta alla rovina. La sua nascita è legata spiritualmente al pentimento di aver creato armi nucleari, ma col tempo si è evoluto cambiando più volte nel corso dei decenni.

Il suo successo ha portato alla nascita dell'Isola dei Mostri e di tanti rivali/amici nel corso del tempo e tra questi c'è Gamera. Un mostro misterioso nato nel 1965 come antagonista; una gigantesca tartaruga zannuta in grado di sputare fuoco e fiamme e di volare. Col tempo è diventata un beniamino dei più piccoli tanto da essere citato in tantissime pubblicità, manga e persino in una vera famiglia di tartarughe marine.

Ora i nostri gargantueschi mostroni si trovano alle prese con un grosso problema. Il malvagio professor Shorizawa li ha ridotti in piccole creaturine paffute e li ha infilati in un mega stabilimento labirinto.

Per poter tornare come prima i due amici/nemici dovranno quindi collaborare e recuperare delle piccole sfere sparse lungo tutti i livelli di gioco entro un limite di tempo. Il solo modo





GIUDIZIO FINALE

» Giocabilità 90%
Semplice e stimolante nel suo apprendimento e modo di gioco. Ci piace molto la collaborazione in game.

» Longevità 90%
Un titolo che si lascia giocare senza troppi patemi e con il grado giusto di difficoltà.

per tornare giganteschi, liberi e felici di distruggere tutto ciò che gli capita a tiro.

Muoversi per il gioco è semplice. Basta raccogliere le palline, evitare i mostri (che possono essere solo storditi), seguire i consigli, raccogliere le password e collaborare. Alcuni livelli potranno essere portati a termine solo se Godzilla si fa aiutare da Gamera e viceversa. Idea simpatica e stimolante.

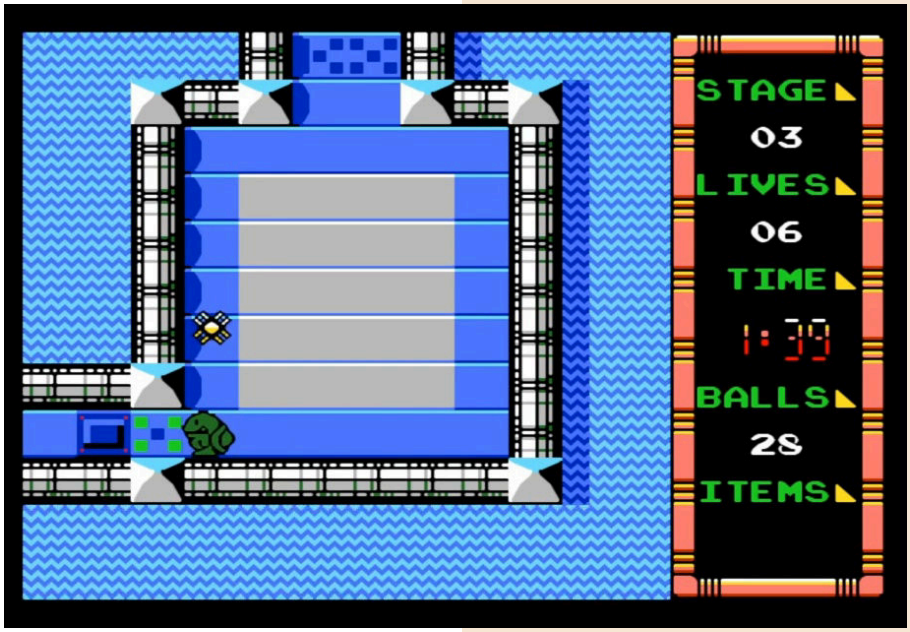
Il titolo è divertente e ben sviluppato. C'è un uso meraviglioso dei colori e gli sprite sono ben caratterizzati e in puro stile "Super Deformed Kawai". Anche la musichetta è ben fatta e piuttosto accattivante.

La modalità di gioco è unica e non sono previsti livelli di difficoltà, consigliamo caldamente di leggere i file del manuale contenuti nella rom del gioco.

Godzilla Balls gira su MSX1, MSX2, 2+ e TR, utilizza 16kb di Ram ed è un Rom da 64 kb. È prevista anche una futura edizione fisica entro la fine dell'anno.

Un bel puzzle game con parti d'azione che merita di essere provato e giocato. Un bel lavoro dei Paxanga Software!

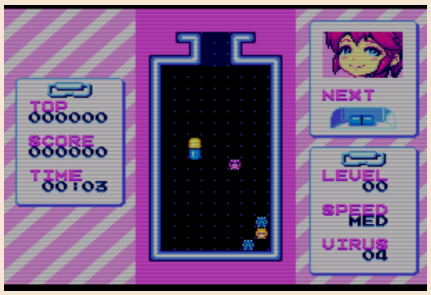
di **Giampaolo Moraschi**

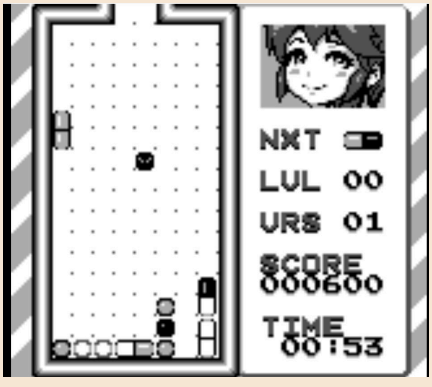


NEW GAME

MAI NURSE

Anno: 2024
Editore/Sviluppatore: Lunoka
Genere: Puzzle game
Piattaforma: Game Boy, Pc Engine Turbo Grafx 16, Sega Game Gear, Sega Master System
Sito: https://itch.io/queue/c/630067/game-boy-games-console-retro-homebrew-gbc?game_id=2603919





Le istruzioni di questo gioco riportano "aiuta Mai a sbarazzarsi dei virus con le pillole, allinea 4 slot dello stesso colore in orizzontale o verticale per distruggerli e progredisci attraverso i livelli di questo classico puzzle game!".
Mi ricorda qualcosa? Ma certo!
È Dr. Mario, icona assoluta dei classici giochi per Game Boy.
Mai Nurse è un clone perfetto in meccaniche di gioco. Stessi virus arcigni, stesso tipo di pillole da allineare.
Cambiano solo la musica e la protagonista.
Basta con quel baffuto idraulico travestito da medico! Ecco la prosperosa e kawai Mai.
Un titolo divertente sviluppato in multi piattaforma e realizzato con il GBDK2020, piattaforma di sviluppo per sm83, Z80 e 6502.
Uno spudorato clone ma sempre divertente e disponibile in futuro anche su Megadrive, Snes, Saturn e GBA.
Lo sviluppatore sta pensando anche di realizzare una versione fisica verso la fine del 2024.
Ho provato sia la versione Gb (in real) che quella PC Engine (in emulazione).
Il gioco è semplice e mantiene lo stesso livello di interesse dell'illustre "padre".
Ho trovato più gradevole la versione Gb, ideale da portare in giro tra un turno o l'altro di lavoro o in università.
Su Pc Engine tutto bello colorato e con un simpatico motivetto d'accompagnamento.
Se vi piaceva Dr. Mario, adorerete questo titolo.

di **Marta Rossman**

GIUDIZIO FINALE

» Giocabilità 90%
Mantiene tutta la semplicità e il game play del titolo originale.

» Longevità 90%
Meno incisivo di Tetris o Puyo Puyo, ma sempre un gran bel titolo da giocare e rigiocare.

NEW GAME

BOMB JACK
THE REMAKE

Anno: 2024
Editore/Sviluppatore: MikGames
Genere: Platform
Piattaforma: Sega Master System
Sito web: https://www.smspower.org/Homebrew/BombJackRemake-SMS

Da piccolina adoravo giocare a questo gioco sul Mame di casa.
Appartengo ad una generazione nata dopo il boom delle sale giochi arcade, ma ne sono sempre stata affascinata. Sono una di quelle che, fin da ragazzina, si fionda nelle poche sale giochi rimaste al mare.
Questa mia passione per i vecchi titoli è merito di mio padre e di chi mi ha cresciuto a pane e “old school”.
Questa versione ad 8bit per Sega Master System di Bomb Jack è… una Bomba!!!
Il gioco rimane avvincente come in sala giochi e possiede anche tutte le fasi di gioco e le caratteristiche dell’originale, cosa che in altri casi non è stato possibile.
C’è tutto. Le fasi di gioco veloci e furiose, adatte a chi ama sfidare il pericolo mentre cerca di spegnere tutte le bombe nelle piattaforme. Ci sono tutti i mostri fetidi e la fluidità del piccolo supereroe rosso e blu.
È possibile settare il livello di difficoltà e altre opzioni prima di cimentarsi con l’avventura.
Penso sia una delle versioni più divertenti di questo titolo, perfetto se caricato sul vostro dispositivo portatile preferito anche in emulazione (io ci gioco in emulazione sullo smart phone).
Una chicca inaspettata nata per partecipare al concorso indetto dalla pagina SMS Power!, punto di riferimento per la viva comunità dedicata al Master System.
Bello!

di **Ingrid Poggiali**

GIUDIZIO FINALE

» Giocabilità 90%
Un titolo dalle dinamiche di gioco semplici ma letale. Bisogna sapersi muovere bene tra i livelli ma ogni livello completato dona un gusto unico.

» Longevità 90%
Gioco da affrontare con lo spirito dei vecchi titoli degli anni passati. Un gettone (start in questo caso) tira l’altro.

NEW GAME

ARKON BLAZE

Anno: 2024
Editore/Sviluppatore: VoxVideogame
Genere: Shoot em up
Piattforma: Commodore 64
Sito web: https://voxvideogame.itch.io/arkon-blaze-c64 - https://voxvideogame.itch.io/

Sensible Software e Commodore sono stati per anni intrecciati assieme.
Un rapporto di coppia fruttuoso nelle due epoche d'oro dei videogiochi su home computer.
Nell'era dei 16 bit avevamo tantissimi titoli amati creati dalla software house inglese, due su tutti sono incisi nella tavola dei comandamenti dei giochi per Amiga: Cannon Fodder e Sensible Soccer.
Ma l'amore nasce lontano, già in piena era 8 bit dove sfornarono alcuni piccoli capolavori come Wizball oppure Microprose Soccer o anche International 3D Tennis.
Tra i loro successi troviamo anche il S.E.U.C.K. ovvero lo Shoot'em up Construction Kit, un programma che permetteva allora ed oggi, la creazione di videogiochi su C64, ma anche su Amiga e Atari ST.
La versione C64 spopolo a tal punto da creare una vera e propria "scena" e tanti giovani e virgulti sviluppatori in erba. La gloriosa Zzap! Italiana del tempo dedicava anche molto spazio ai titoli realizzati con il SEUCK.
Siamo nel 2024 ed il programma conta sul groppone ben 37 anni (uscì infatti nel 1987), ma sono ancora tanti gli sviluppatori che lo utilizzano per le proprie opere.
Tra questi troviamo Pinov e la sua Voxvideogame, che sviluppa a spron battuto per il C64.
Oggi infatti diamo un'occhiata all'ultimo nato in casa Voxvideogame:

Arkon Blaze.
La storia parte in un'altra dimensione, in un mondo chiamato Arkon. Questo pianeta fu creato da un malvagio essere chiamato "L'Oscuro Signore", che scoprì l'essenza e il segreto dell'Arkognosim, una fonte di energia presente sulla terra abitata dai docili e pacifici "gnosim", una popolazione gnomica.
Queste piccole creature servono

all'Oscuro Signore come fonte di energia, ed incarica i suoi demoni di rapirli.
Un ribelle si oppone alla volontà malvagia e si getta a capofitto nell'impresa di salvare tutti i piccoli abitanti del pianeta e sconfiggere il malvagio despota.
Lo scopo è quindi semplice: combattere i mostri, salvare i piccoli "gnosim", raccogliere dei pesci energetici bonus (punti extra e vite) e raggiungere l'ultimo livello per distruggere il Big Boss.
Impresa sulla carta veloce ma ovviamente non semplice. Nessun gioco è semplice e ovviamente i nemici cercheranno di farci la pelle.
Ma com'è Arkon Blaze nel suo insieme? Un prodotto budget realizzato con volontà. Mi ha ricordato, vagamente, Dragon Spirit per stile e ambientazione ma senza tutti i power up presenti.
Il nostro eroe si muove bene ed è animato con cura.
Così come sono animati con cura i vari mostri e gli sprite "amici" presenti sullo schermo.
Forse avrei azzardato di più sulle ondate di assalto nemiche, ma quello che c'è funziona.
Dal punto di vista grafico c'è da dire che mi sarei spinto un po' di più sulla scelta dei colori. Alcune volte si confondono gli oggetti sullo sfondo e risulta tutto confuso, ma non per questo meno giocabile.
Effetti sonori dignitosi.
Livello di difficoltà bilanciato anche se, come dicevo, mi sarei azzardato ad inserire qualche elemento in più delle dinamiche in gioco.
Power up o cose di questo genere. Solo punti extra e vite extra sono un po' pochino. Avrei aggiunto armi differenti o una smart bomb per esempio.
Tirando le somme di fine giornata però mi sento di promuovere il gioco per lo sforzo tecnico e la struttura. Magari una versione "Enhanced" in futuro setterà lo stile e farà aumentare il voto definitivo.
Lo trovate sul sito che vi ho messo nel commento di descrizione al prezzo di 3.49 dollari, una sciocchezza ;-).

di **Carlo Nithaiah Del Mar Pirazzini**

GIUDIZIO FINALE

» Giocabilità 70%
Avrei osato di più ma quello che c'è va bene.

» Longevità 75%
Non tanti mostri sullo schermo ma la difficoltà è ben strutturata.

BEACH HEAD

Anno: 1983
Editore: Access Software
Genere: Shoot'em up
Piattaforma: Commodore 64

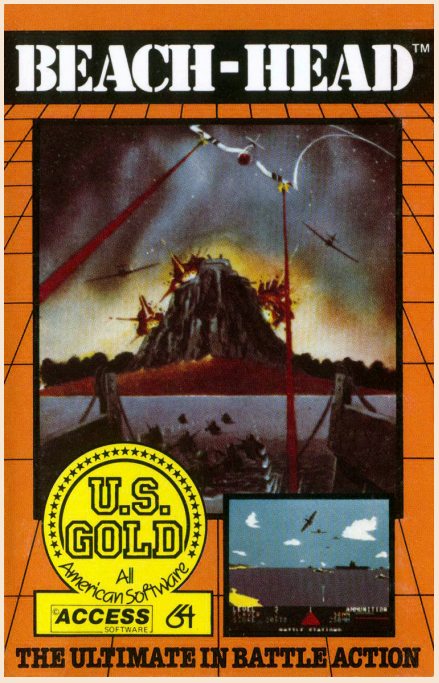
Con tutto il rispetto per la situazione palestinese ed anche per le guerre passate con relative vittime, presentiamo un gioco dove la guerra fortunatamente si fa per finta (e in confidenza vi posso dire che è l'unica guerra divertente, perché simulata). In più l'unica vittima potrebbe essere solo il joystick causa uso spropositato o la nostra vita sociale se ne abbiamo una.

Beach head della statunitense Acces Software è un gioco strutturato su cinque livelli come piacciono a me, ovvero modalità di gioco diverse. Ciascuna con un incarico ben specifico che andrà a condizionare i successivi a seconda della nostra bravura e strategia militare.

Visto il periodo in cui è stato pubblicato (dal 1983 al 1985) si presuppone che l'ambientazione fosse la seconda guerra mondiale, nel periodo degli sbarchi alleati, visto il titolo.
Partiamo subito con con il primo livello, opzionale, nel senso che non è obbligatorio superarlo, anche se lo consiglio vivamente in modo da vincere più facilmente nel successivo: qui è coinvolta la nostra nave in una planimetria in cui sono visibili siluri e mine che ci impediranno di raggiungere la sponda opposta.
Il secondo livello è uno sparatutto a schermata fissa in mare aperto in cui prenderemo il comando di una torretta contraerea situata su una nave e dovremo abbattere un certo numero di aerei nemici che ci vengono incontro per farci diventare un relitto.
Il terzo è sempre a schermata fissa molto simile al precedente ma questa volta dovremo abbattere navi nemiche con una enorme precisione per regolare l'altezza giusta del cannone per centrare il bersaglio battendolo sul tempo, pena portarci a casa uno o più siluri nemici.
Il quarto livello è simile al primo ma questa volta alla guida di un bel carro armato in un livello a scorrimento da destra verso sinistra affrontando vari nemici e ostacoli degni di un terreno di guerra.
Il quinto e ultimo livello dove finalmente raggiungeremo la nostra collina, beh, cercate di godervelo e di trovare da soli il modo di far sventolare bandiera bianca, altrimenti qui tutto diventa una soluzione se non uno spoiler. Posso dirvi solo di non arrivare con un solo carro armato, altrimenti nemmeno se siete dei generali pluridecorati in pensione riuscirete a portarlo a termine. Beach head a livello di giocabilità non l'ho trovato per nulla noioso anche se i livelli, seppur diversi tra loro, hanno qualche similitudine.

Per quanto riguarda la longevità ci ho messo più del servizio militare obbligatorio ad arrivare all'ultimo livello (anche con dieci vite), quindi c'è di che divertirsi e la compagnia è assicurata. Il sonoro purtroppo è assente, anche se mi sembra un fatto più che giustificabile in un gioco di guerra.
Il gioco ebbe abbastanza successo per poterne fare un seguito, Beach head 2 e come se non bastasse la serie ispirò alcuni giochi usciti per PC negli anni 2000.
Chiudo sperando che in tutto il mondo cessino le ostilità e comincino competizioni più goliardiche, amichevoli e/o video-ludiche come i giochi che si trovano nei negozi, su internet e… in questa rivista!

di **Daniele Brahimi**



GIUDIZIO FINALE

» Giocabilità 70%
Dieci vite sembrano tante per superare cinque livelli, ma…

» Longevità 750
Abbastanza semplice, anche se richiede troppa precisione in alcuni punti.

Preservare… per vivere il futuro

I videogame sono uno dei prodotti culturali più importanti della nostra epoca e, se anno dopo anno le esperienze di gioco si fanno sempre più complesse, interattive e aggiornate, la conservazione sarà invece una delle sfide più complesse del nostro futuro.

In un recente studio, la Video Game History Foundation ha rivelato una verità sconvolgente per tutti noi appassionati: l'87% dei videogiochi rilasciati prima del 2010 sono "in pericolo critico". Parliamo di qualcosa come 9 giochi su 10 che potrebbero diventare inaccessibili alle future generazioni. Ma perché potremmo perdere questo pezzo di vita/storia? La risposta è più complicata del previsto.

I videogame per loro natura hanno un ciclo vita. Spariscono dai negozi, dagli scaffali e diventano introvabili. Solo le saghe più famose vengono riproposte, ma migliaia di titoli spariscono senza lasciare una traccia fisica. Sul banco dei colpevoli troviamo anche la legge sul copyright che impedisce alle biblioteche e alle istituzioni culturali di preservare, mostrare e condividere i videogiochi allo stesso modo di film, libri o musica.

Tutto questo, assieme al mercato lievitato sospinto in alto dai retro-collezionisti e alla scomparsa fisica dei supporti, potrebbe trasformarsi in una perdita culturale incalcolabile. Eppure l'industria videoludica al momento vale circa 180 miliardi di euro e supera sia quella cinematografica sia quella musicale! E allora perché succede tutto questo?

Questo 2024 potrebbe essere l'anno cruciale. Negli USA è prevista una discussione sulle leggi sul copyright, anche se il tempo scorre veloce.
Occorre qualcosa di forte, una mozione pubblica, un'iniziativa, qualcosa che modifichi le leggi attuali. Serve che si riconoscano l'emulazione e l'implementazione come fonti di "preservazione storica". Occorre che si permetta alle associazioni di "dumpare" i titoli dei gloriosi home computer del passato senza aver ripercussioni in merito.

Non si tratta di pirateria ma di libertà e di storia.
Perché tra tutte le avventure affrontate in questi 40 e rotti anni si rischia di non poter premere più il tasto "fire" e che sullo schermo appaia la scritta "GAME OVER". Viva la preservazione!

Carlo Nithaiah Del Mar Pirazzini

Disclaimer

RetroMagazine World (fanzine aperiodica) è un progetto interamente no profit e fuori da qualsiasi circuito commerciale. Tutto il materiale contenuto è prodotto dai rispettivi autori e pubblicato grazie alla loro autorizzazione.

RetroMagazine World viene concessa al pubblico con licenza: Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale (CC BY-NC-SA 4.0 INT) https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it

In pratica sei libero di: condividere, riprodurre, distribuire, comunicare o esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato, modificare, rielaborare, trasformare il contenuto e basarti su di esso per altre opere, alle seguenti condizioni:

Attribuzione
Devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi farlo in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o l'utilizzo del materiale da parte tua.

NonCommerciale
Non puoi utilizzare il materiale per scopi commerciali.

StessaLicenza
Se rielabori, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi con la stessa licenza del materiale originario.

Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

Divieto di restrizioni aggiuntive
Non puoi applicare termini legali o misure tecnologiche che impongano ad altri soggetti dei vincoli giuridici su quanto la licenza consente loro di fare.

RetroMagazine World
Anno 8 - Numero 47 - APRILE 2024

Direttore Responsabile
Francesco Fiorentini
Vice Direttore
Marco Pistorio
Coordinatore Redazione/Editing
David La Monaca
Responsabili Area Web / Social Media
Giorgio Balestrieri / Carlo Nithaiah Del Mar Pirazzini